Modelo predictivo para Start-Up Agrícola

Sergio Castelblanco, José Sandoval, Aleksey Sepúlveda

INTRODUCCIÓN

En el presente artículo se analiza el problema al que se enfrenta una Start-Up del sector agrícola frente a la estimación de la demanda, cuyas imprecisiones pueden originar niveles de sobre producción, riesgos en la generación de volúmenes indeseables de inventarios y, por lo tanto, desencadenar en una elevación de costos. En este artículo se propone un conjunto de modelos de predicción de demanda y precio, que permita brindarle a la organización un sistema de gestión y predicción para planear, con la debida anticipación, la demanda esperada y así gestionar los inventarios de manera inteligente, garantizando un proceso logístico más flexible y efectivo.

La predicción de la demanda y el precio de los productos es un desafío en el sector agrícola, ya que las ventas presentan comportamientos sujetos a evoluciones cíclicas y estacionalidades propias de la forma de producción y consumo de este tipo de productos. La información con la que se trabajó evidencia estos comportamientos cíclicos. La base de datos suministrada cuenta con 121 productos de distintas clases entre ellas frutas, vegetales, granos etc. Las series de tiempo de estos productos presentan fluctuaciones importantes en las cantidades demandadas que incluyen periodos de cero cantidades vendidas y picos de solicitudes en algunos casos, así como precios fluctuantes y dispersos en algunas categorías de productos, entre otras características naturales en este tipo de sector.

El reto analítico que se plantea es el no contar con información uniforme (comportamientos de compra similares entre productos), ni tener continuidad en la totalidad de la serie (información diaria) para el periodo de tiempo de la base de datos suministrada. Adicionalmente, se presentan algunos productos con información escasa en demanda y precio lo cual hace más compleja la estimación de un modelo global y el uso de modelos tradicionales de series de tiempo bajo metodologías de Box and Jenkins y/o métodos de descomposición y suavización exponencial, que no permiten procedimientos de modelaje eficientes para tal reto.

Luego del análisis exploratorio, una de las alternativas más robustas y simples para reducir la dimensionalidad del problema fue implementar el principio de Pareto, también conocido como la regla 80-20 la cual postula que el 20% de las causas explican el 80% de los problemas, en este caso, el 20% de los productos representan el 80% de los ingresos. Este principio permitió pasar de 121 productos a 27 productos (22%) que explican el 80% de los ingresos totales de la Star-Up. Estos mismos productos tienen una alta frecuencia, aproximadamente en el 95% de los casos, con los productos de mayor nivel de demanda.

El enfoque metodológico adoptado para lograr los resultados propuestos se definió en 6 etapas: entendimiento del problema, análisis exploratorio y reducción de dimensionalidad, estimación e imputación de datos, transformación secuencial de datos, modelamiento predictivo, selección del modelo y análisis de resultados. En los últimos pasos se presentan tres enfoques de modelamiento, basados tanto en técnicas tradicionales como en métodos supervisados de redes neuronales para la predicción de la demanda. Es importante resaltar que los modelos construidos trabajan de manera independiente para estimar demanda y precio.

En primera instancia, se utilizó el método clásico de previsión para series de tiempo intermitentes propuesto por Croston (Croston, 1972) el cual utiliza como base la atenuación exponencial simple, la cual separa la serie en dos partes: una serie con valores positivos de demanda, y la segunda, con los tiempos entre demandas consecutivas no vacíos. En cada una de esas series se estima la previsión por medio de suavización exponencial y luego, ambos valores son actualizados cuando existe un valor no nulo de demanda. Con ambas previsiones se estima un resultado en términos de la relación demanda/ periodo.

Un segundo modelo propuesto utiliza un modelo de redes neuronales recurrentes Long Short Term Memory LSTM (Schmidhuber, 1997) que son muy usadas actualmente para la predicción de series temporales dado que tienen bloques de memoria que están conectados a través de capas. Estos bloques de memoria facilitan la tarea de recordar valores para largos o cortos períodos de tiempo y pueden ser usados para aprender los rezagos y predecir pasos adelante. Un tercer método aplicado fue el de redes neuronales convolucionales dilatadas causales cuyo enfoque está inspirado en el trabajo desarrollado por google en 2016 en su modelo WaveNet1 (Van den Oord, 2016). Este método, en una versión más simplificada, nos permite extraer patrones y detalles de la serie, usando su capacidad de convolucion y usando información del pasado para predecir el futuro, a partir de su capacidad de configuración causal.

El presente artículo se distribuye de la siguiente manera, en la primera parte se hace una breve descripción de los datos, mostrando el comportamiento heterogéneo antes descrito y el Pareto definido desde el punto de vista de los ingresos generados. En la segunda parte se hace referencia al procedimiento de estimación e imputación de datos faltantes. En la tercera sección se presenta la aplicación de los modelos de predicción elegidos, iniciando por el método clásico de previsión para series de tiempo intermitentes, luego RNN LSTM y finalmente redes convolucionales dilatadas causales. En la parte final se presentan la comparación del desempeño de los algoritmos, sus resultados y conclusiones Finales

Teniendo en cuenta que se desarrollaron algoritmos en los programas R y Python, se alternan diferentes salidas de los mismos, de acuerdo con los modelos aplicados en cada instancia.

Palabras clave: CROST, IDCLASS, TSB, LSTM, Convolucion1D, Redes Neuronales, Deep Learning, Analytics.

[1] https://arxiv.org/pdf/1609.03499.pdf

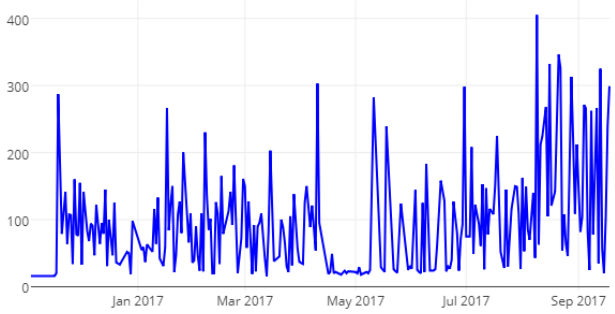
1. ANÁLISIS DESCRIPTIVO

La base de datos está conformada por información de demanda para 121 productos agrícolas, en un periodo de tiempo que abarca desde el 3 de noviembre de 2016 hasta el 17 de septiembre de 2017:

	Cliente	Fecha	Pedido	Precio	Producto	Nombre_producto
0	Cliente26	18/09/2017	20	700	VER0049	Yerbabuena / 100 gramos
1	Cliente26	18/09/2017	10	1200	FRU0024	Limón Tahití / Libra
2	Cliente26	18/09/2017	4	1600	VER0041	Puerro / Libra
3	Cliente26	18/09/2017	10	1500	VER0038	Pimentón Rojo / Libra
4	Cliente26	18/09/2017	8	1100	VER0010	Zucchini Verde / Libra
5	Cliente26	18/09/2017	30	1000	VER0052	Plátano Maduro / Libra
6	Cliente26	18/09/2017	24	1000	TUB0012	Zanahoria / Libra
7	Cliente3	18/09/2017	1	4500	VER0050	Rúgula / Libra
8	Cliente3	18/09/2017	1	3500	VER0043	Repollo Morado / Unidad
9	Cliente3	18/09/2017	2	1500	VER0038	Pimentón Rojo / Libra
10	Cliente3	18/09/2017	2	1400	VER0028	Lechuga Crespa / Unidad
11	Cliente3	18/09/2017	1	1400	VER0020	Espinaca / Libra
12	Cliente3	18/09/2017	2	500	VER0017	Cilantro / 100 gramos
13	Cliente3	18/09/2017	3	7000	VER0016	Champiñón / Libra
14	Cliente3	18/09/2017	1	1600	VER0013	Cebolla Cabezona Roja / Libra

Tabla 1. Base de datos Start-Up Productos agricolas

A partir de las variables “Pedido” y “Precio”, se determinó una nueva variable que agrupa ambos comportamientos: el ingreso. A continuación, se observa el comportamiento del ingreso en el tiempo, de manera agregada (total ingreso de la Start-Up).

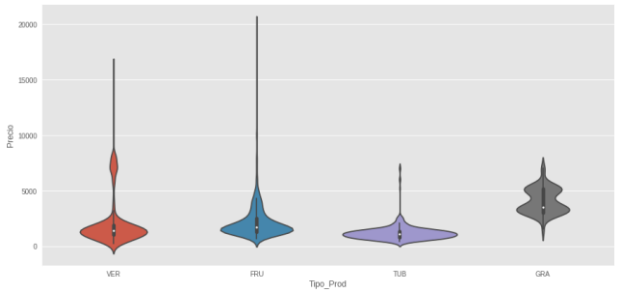


Gráfica 1. Ingreso en el tiempo

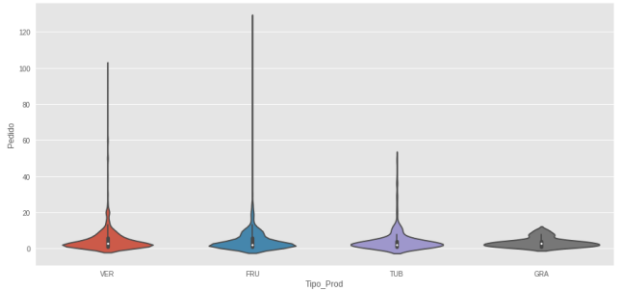
Se aprecia una serie de tiempo con una variabilidad importante, estacionaria respecto a su media, pero con una varianza heterogénea hacia el final del periodo observado, en este sentido, la aplicación del enfoque Box Jenkings requeriría la transformación de la serie.

Ahora bien, con el fin de observar el comportamiento de la demanda de manera desagregada, en primera instancia se analizaron los datos a partir de la agrupación natural de los productos en cuatro categorías: verduras (VER), Frutas (FRU), Tubérculos (TUB) y Granos (GRA).

Se identifica el comportamiento de las variables de precio y pedido para los diferentes grupos, revisando en cada uno de estos el comportamiento de las variables precio y cantidad:



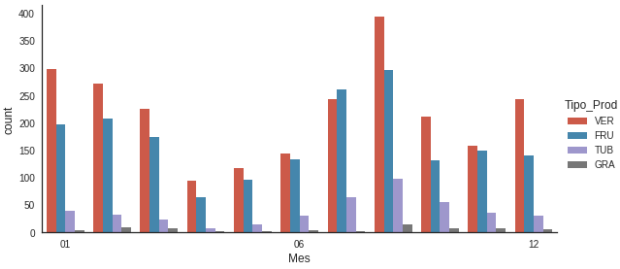
Gráfica 2. Categorias producto en Precio



Gráfica 3. Categorias producto en Pedido

Se observa una dispersión importante tanto en el precio como en el número de pedidos para el caso de las categorías verduras, frutas y tubérculos, mientras el grupo de granos parece más centrado en ambas variables.

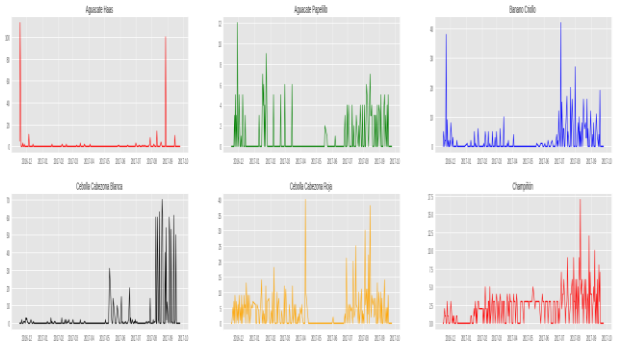
Así mismo, se revisó el comportamiento temporal de las cantidades demandas en las categorías mencionadas:



Gráfica 4. Demanda en Categoria productos

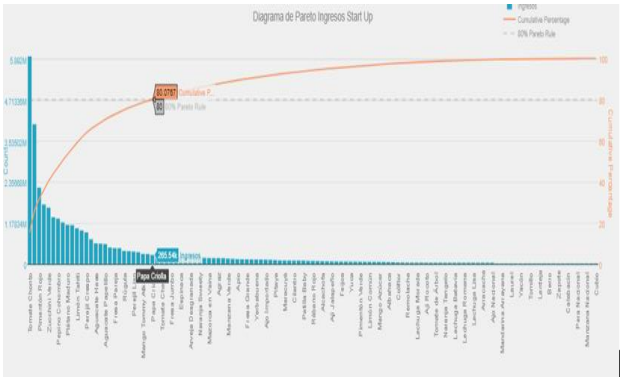
Si bien no se cuenta con información para un periodo completo (un año), se aprecia cierta estacionalidad en la demanda de los grupos, en la que se tiene un pico de pedidos en el mes de junio de 2017, para todas las categorías. Nuevamente la categoría de granos presenta la menor variación temporal.

Teniendo en cuenta esta dispersión y variabilidad temporal en los grupos o categorías, a continuación, se muestra la revisión de las series de demanda por producto (la grafica muestra algunos productos)



Gráfica 5. Demanda 12 Productos.

Para productos dentro de un mismo grupo, se observa una variación importante en el comportamiento, tanto en el precio como en las cantidades demandadas, de manera que se identifica la necesidad de modelar la predicción de demanda para cada uno de los productos, no obstante, luego de realizar un análisis del ingreso de la Start-Up, se encontró que 27 productos representan el 80% de los ingresos de la compañía:



Gráfica 6. Participación acumulada de Ingreso en los productos

2. ESTIMACIÓN E IMPUTACIÓN DE DATOS

Una vez reducida la dimensionalidad del problema, es decir, definido los productos Pareto a predecir en los modelos, se debe iniciar uno de los procesos más importantes del análisis, el cual involucra la primera versión de la base de datos para ser usada en los modelos futuros. Para esto se realiza una agrupación por fecha, producto, demanda y precio que permita establecer las series de tiempo en una única fecha de tiempo continuo.

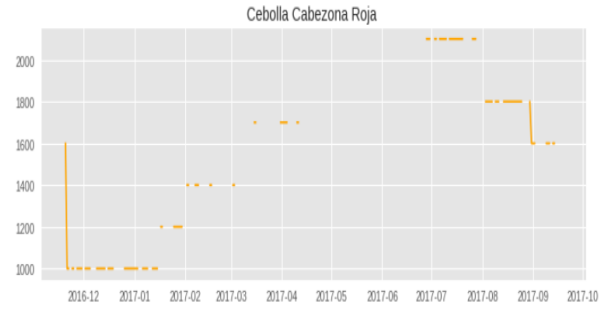
Fecha	Acelga	Agraz	Aguacate Haas	Aguacate Papelillo	Al
2016-11-03	0	0	0	0	
2016-11-16	0	0	0	0	
2016-11-17	0	0	5	0	
2016-11-18	0	10	113	0	
2016-11-20	0	0	0	0	
2016-11-21	0	0	0	0	
2016-11-22	0	0	0	0	
2016-11-23	10	2	3	3	

Tabla 2. Tabla transpuesta de información

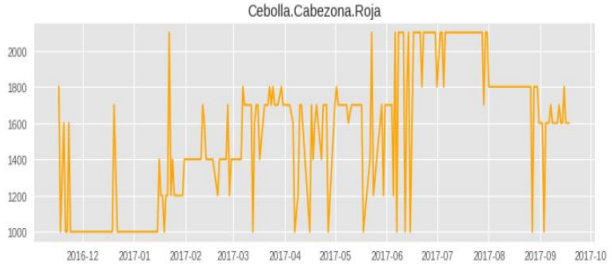
Una vez realizado este procedimiento, se observa un gran porcentaje de valores faltantes, lo cual se convierte en otro de los retos del problema, dado que para el caso del precio, la intermitencia de la serie se da en grandes proporciones y no hay un camino muy claro para la estimación de estos datos en los días faltantes,

Gráfica 7. Missmap del precio

En el grafico anterior se observa el missmap2 para la variable precio, evidenciando entre el 60% y 70% de datos faltantes en algunos productos, siendo este un problema que corresponde a la categoria de MAR (Missing At Random)3. Para abordar este problema se probaron múltiples métodos, entre ellos imputación con la media, imputación con el precio del día anterior y la construcción de un modelo predictivo que permitan suavizar la serie de cada producto. Este último método resultó ser más eficiente, dado que no se agregaban sesgos basados en datos arbitrarios si no que, de acuerdo con la estructura misma de los datos, se usó un modelo predictivo de Random Forest (Ho, 1995) usando la librería mice (Multivariate Imputation by Chained Equations) en R, que estimó los datos faltantes de la serie. Los resultados fueron satisfactorios, un ejemplo a continuación:

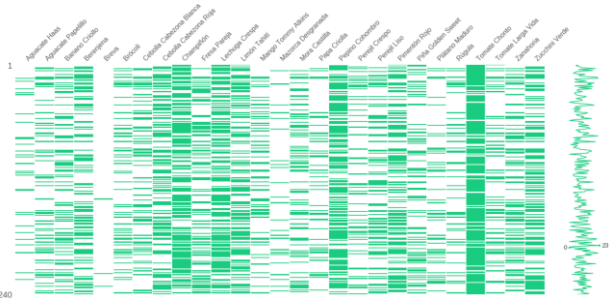


Grafica 7. Serie Intermitente Cebolla Cabezona Roja.



Grafica 8. Serie Resultado Ramdon Forest.

Para el caso de la demanda, el procedimiento lógico fue más sencillo dado que se tomó como supuesto que los días de missing data no se presentó venta del producto y se imputa con valores de 0.



Grafica 9. Missmap de la variable demanda.

En el grafico anterior se observa el missmap para la variable demanda, evidenciando entre el 62% y 75% de datos faltantes en algunos productos.

Una vez estimados e imputados los datos para ambos casos, Demanda y Precio, se procede a la unificacion de las bases de datos para la construccion de los modelos predictivos.

3. MODELOS DE PREDICCIÓN APLICADOS

Antes de la explicación técnica de cada modelo, es importante comentar los principales retos y obstáculos enfrentados en el proceso de modelamiento. En ese sentido, se exponen algunos cuestionamientos que se tuvieron que resolver sobre la marcha del desarrollo de este trabajo, ¿Cómo transformar la base de datos de tal manera que pueda ser interpretado como un método supervisado?, ¿Cómo construir un modelo predictivo multirespuesta para series de tiempo?, ¿Con que lógica se puede dividir la base de datos train and test?, ¿Que métrica idónea definir para evaluar y seleccionar los modelos?.

Ante estas incógnitas se decide abordar el problema desde los enfoques más sencillos, hasta los modelos más complejos, comparando sus diferencias en eficiencia y desempeño. Las respuestas a los cuestionamientos planteados se exponen a continuación.

3.1 Ts Intermitente Método De Croston

El método de Croston (Croston, 1972) se basa en la demanda que aparece ocasionalmente; es decir, en donde ciertos periodos de tiempo no presentan demanda alguna o valores cero, y cuando ocurre,

[2] Gráfica que permite identificar los registros que se encuentran vacíos dentro de una tabla de datos.

[3] Significa que no existe relación entre la falta de datos y los valores, observados o perdidos. Esos puntos de datos faltantes son un subconjunto aleatorio de los datos.

su tamaño no es homogéneo, por lo cual es muy difícil de predecir.

Este método separa los componentes de la demanda y del modelo de forma independiente en dos estimados: intervalos de tiempo entre observaciones y en cantidad, y a partir de ello logra predecir separadamente el valor de la demanda no nula y el tiempo entre arribos de las cantidades sucesivas no nulas, utilizando el suavizamiento exponencial simple (Croston, 1972). Cómo consideración es importante indicar que, si el método se usa en una serie de tiempo no intermitente, es decir, sin valores ceros, entonces el método es idéntico al de suavizamiento exponencial simple (Croston, 1972).

En términos matemáticos y sencillos:
dt: demanda en el periodo t.

$\hat{D}_t$: Previsión de la demanda no nula para el período.

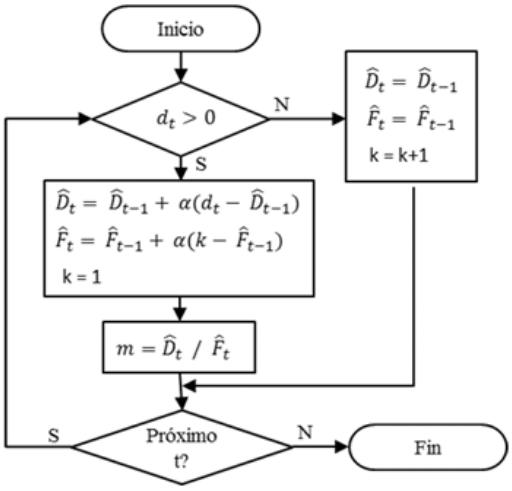
ft: tiempo entre dos demandas no nulas;

$\hat{F}_t$: Previsión del intervalo de demanda;

k: intervalo desde la última demanda no nula;

α: parámetro de atenuación, 0≤α≤1;

m: demanda promedio en el período.

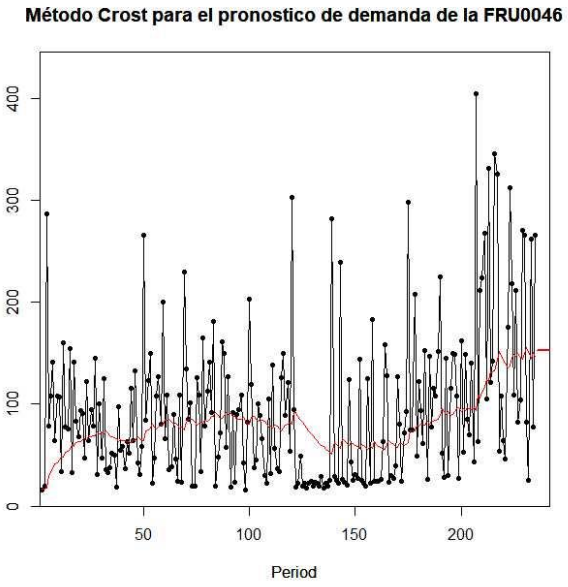


Fuente: (Santa Cruz R. & Correa, 2017)

Teniendo en cuenta lo anterior el resultado es:

m= $\hat{D}_t / \hat{F}_t$

Resultados del modelo:



Grafica 10. Resultados del modelo TS Intermitent.

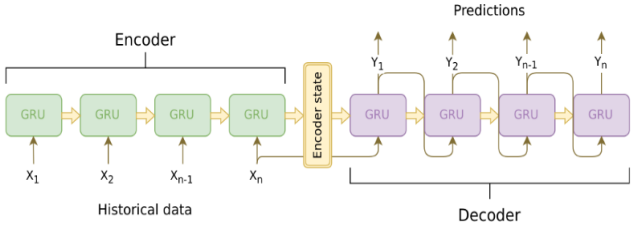
Se observa un pobre desempeño respecto a la precisión del modelo, ya que este en ultimas está suavizando un promedio de la serie, sin recoger de manera adecuada el comportamiento de esta.

3.2 Transformación de los datos método sequence to sequence.

Antes de continuar con los modelos de redes neuronales es importante hacer una pausa y

detenerse a pensar cómo resolver uno de los retos más importantes del problema a resolver y es el de transformar las series de tiempo a un método supervisado, donde la estructura de los datos debe estar en condiciones y dimensiones adecuadas, de acuerdo con los requerimientos de la arquitectura LSTM y la arquitectura convolucional, es decir, capas de entrada según se necesita, con un input de 3 dimensiones (Samples, Time_Step, Features). Para resolver este problema, se encontraron varios enfoques como por ejemplo usar funciones Shift, que se pueden utilizar para crear copias de columnas con un paso adelante, para emular el problema supervisado. También se exploró el método de seq2seq que es el corazón de los modelos que se explicaran a continuación.

El aprendizaje de seq2seq (Britz Goldie & Luong, 2017), en su núcleo, usa redes neuronales recurrentes para mapear secuencias de entrada de longitud variable a secuencias de salida de longitud variable. Si bien es relativamente nuevo, el enfoque seq2seq ha logrado resultados de vanguardia no solo en su aplicación original traducción automática si no en series de tiempo dado dada su capacidad de predecir una secuencia de menor, igual o mayor a la secuencia de entrada. A continuación un esquema conceptual del modelo:



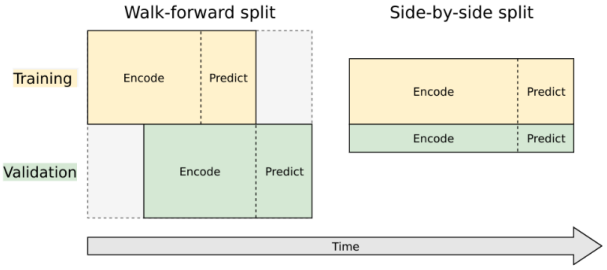
Fuente: (Artur Suilin 2017)

Para la implementación de este modelo se resolvió previamente la forma en como particionar adecuadamente las series de tiempo en intervalos de codificación y decodificación (predicción) a los datos de entrenamiento y validación.

Hay dos formas de dividir las series temporales en conjuntos de datos de entrenamiento y validación:

Walk-forward Split: entrenamos el conjunto de datos completo y validamos en el conjunto de datos completo, utilizando diferentes marcos de tiempo. El plazo para la validación se desplaza hacia adelante en un intervalo de predicción relativo al marco de tiempo para el entrenamiento.

Side-by-side Split: Este es el modelo de división tradicional para el aprendizaje automático convencional. El conjunto de datos se divide en partes independientes, una parte utilizada estrictamente para el entrenamiento y otra parte utilizada estrictamente para la validación.



Fuente: (Artur Suilin 2017)

Se probaron ambos métodos, pero para los modelos finales se usó Walk-forward dado que este permite predecir valores futuros utilizando valores históricos. Para este caso el conjunto de validación abarca el mismo rango de tiempo que el conjunto de entrenamiento, pero se desplaza hacia adelante en el tiempo (en este caso por 1 ó 7 días pronostico solicitado). De esta manera, simulamos cómo el modelo funcionará en datos no vistos que vienen en el futuro.

Posteriormente se implementan las funciones para transformar todas las series, aquí suavizamos la escala tomando log1p y restándole a cada serie la media de la serie del codificador (encoder), luego rediseñamos al formato del tensor (samples, timesteps, features) que las arquitecturas de redes neuronales esperaran.

Se debe tener en cuenta que, para generar predicciones en escalas originales en lugar de escalas codificadas, se debe aplicar una transformación inversa en el momento de la predicción.

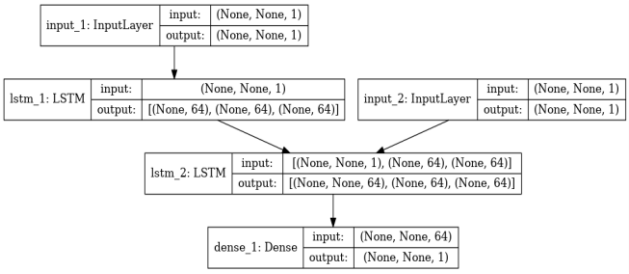
Finalmente, las series de tiempo quedan en el formato y dimensiones adecuadas para la implementación de los algoritmos LSTM y convolucion1D.

3.3 Redes Neuronales Recurrentes LSTM (Long Short Term Memory)

La red LSTM fue que creada en 1997 por Hochreiter y Schimdhuber sin embargo, su popularidad como arquitectura RNN ha crecido en los últimos años para diferentes aplicaciones, entre ellas reconocimiento de voz y series de tiempo (M. Tim Jones 2017), dos tipos de modelos causales que requieren de un pasado para explicar el futuro.

La LSTM se desvió de las arquitecturas de red neural típicas, basadas en neuronas, en cambió presentó el concepto de una celda de memoria la cual puede retener su valor durante un periodo de tiempo corto o largo como una función de sus entradas, lo que permite a la celda recordar lo que es importante y no solamente el último valor que calculó (M. Tim Jones 2017), dadas estas bondades, se decidió construir una red neuronal con este mecanismo para abordar el problema de predicción de demanda y precio.

La arquitectura propuesta para este modelo se diseñó en la herramienta Python bajo el framework más usado en Deep learning Tensorflow (google 2015) junto con la capa de abstracción Keras (MIT 2015).

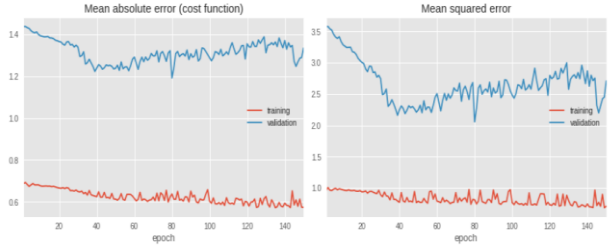


Grafica 11. Arquitectura LSTM - Demanda.

Se observan una capa LSTM de 64 neuronas que permite la codificacion y se concatena con otra capa LSTM que permita la decodificacion de los datos, posteriormente pasa por una capa densa para generar la predicción. Luego de un proceso de calibración de hiperparametros se usó el optimizador Adam, con learning rate de 0.0001 y función de perdida MAE (Mean Absolute Error) y métrica de evaluación MSE (Mean Square Error) dado que nos permite hablar en las mismas unidades de las variables a predecir.

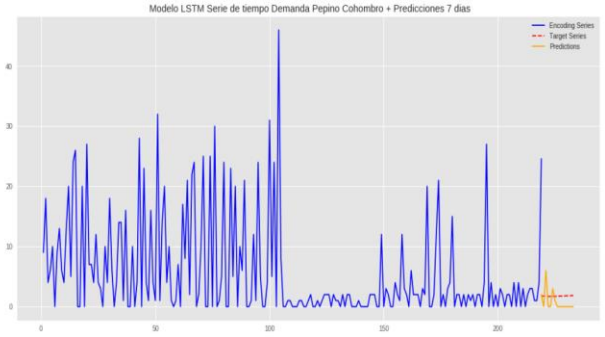
El proceso de entrenamiento se puede observar a continuación:

Numero de épocas 150, Batch Size 1



Grafica 12. Funcion de costo y de error.

Para visualizar el desempeño de la red se muestra un ejemplo para el caso del pepino cohombro:



Grafica 13. Serie de Tiempo Pepino Cohombro y prediccion 7 días.

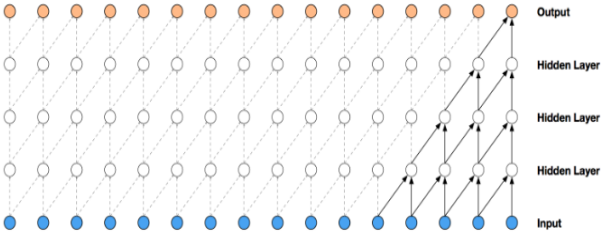
El error del modelo en validacion es de 2.3.

3.4 Redes neuronales convolucionales causales dilatadas

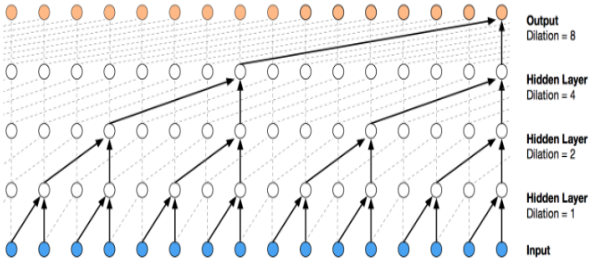
El segundo enfoque desarrollado se basa en la aplicación de convoluciones 1D, de igual manera usando la transformación de datos seq2seq, explicada anteriormente como método de pre-procesamiento para entregar los datos en las dimensiones propias que necesita la red convolucional 1D.

La arquitectura convolucional presentada en este artículo es una versión simplificada del modelo WaveNet diseñado por google como un modelo generativo para audio (en particular, para aplicaciones de texto a voz). El modelo de red de onda se puede abstraer más allá del audio para aplicarlo a cualquier problema de pronóstico de series temporales, proporcionando una estructura lógica y suave para capturar dependencias a largo plazo, sin una cantidad excesiva de ponderaciones aprendidas.

Una convolucion dilatada (también llamada convolucion con agujeros) es una convolucion donde el filtro se aplica sobre un área más grande que su longitud omitiendo los valores de entrada con un cierto paso. Es equivalente a una convolucion con un filtro dilatándolo con ceros, pero es significativamente más eficiente. Una convolucion dilatada permite efectivamente que la red funcione en una escala más grande que una convolucion normal (Deepmind 2016). Es decir aplicar convoluciones simples a series de tiempo haría que el modelo fuera demasiado complejo computacional y estadísticamente. Por lo tanto con convoluciones causales dilatadas tenemos la herramienta adecuada para manejar el flujo temporal, Con una serie de tiempo que se extiende por un año.

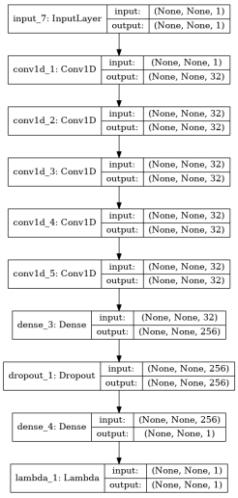


Grafica 14. Visualizacion de convoluciones causales Fuente(Deepmind 2016)



Grafica 15. Visualizacion de convoluciones causales dilatadas
Fuente(Deepmind 2016)

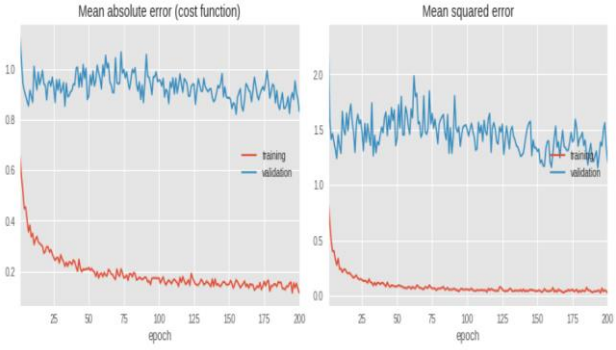
La arquitectura del modelo propuesto para predicción de demanda se presenta a continuación:



Luego de un proceso de calibración de hiperparametros se usaron 5 capas convolucionales para extraer patrones aproximadamente de 8 meses atrás, 32 filtros de ancho 2 por capa, tasa de dilatación exponencial (1, 2, 4, 8, ..., 256), Dropout de 0.2 y funciones de activación Relu y Linear para la capa de salida; se usó el optimizador nadam, con learning rate de 0.001 y función de perdida MAE (Mean Absolute Error) y métrica de evaluación MSE (Mean Square Error) dado que nos permite hablar en las mismas unidades de las variables a predecir. Los parámetros a entrenar fueron de 17,121, lo cual indica la efectividad de este tipo de modelos con bajo costo computacional pero gran capacidad predictiva.

Entrenamiento red neuronal para demanda:

Numero de épocas 200, Batch Size 1

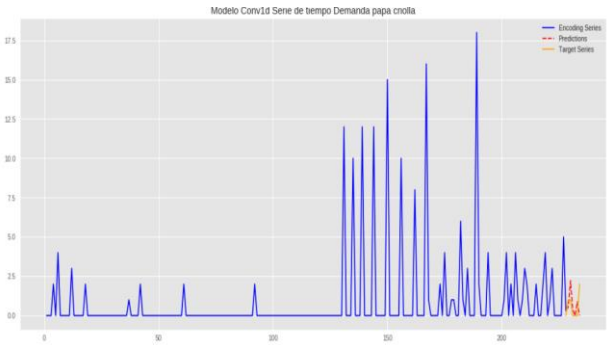


Grafica 16. Funcion de costo y de error.

Se observa que el error promedio es de 1.4 y 1.3 mucho menor que con la red LSTM además el

entrenamiento tiene una curva exponencial decreciente lo cual da sentido al proceso de aprendizaje.

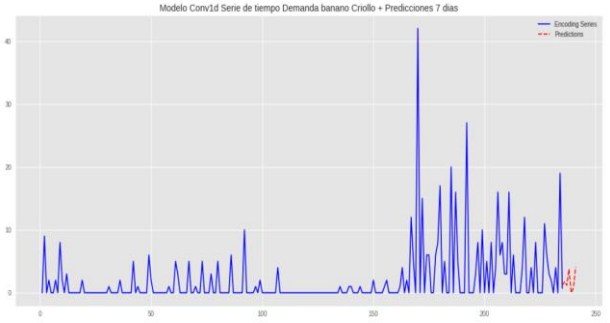
Los resultados fueron satisfactorios. Para visualizar el desempeño de la red se muestra un ejemplo para el caso del papa criolla:



Grafica 17. Serie Papa Criolla y prediccion día.

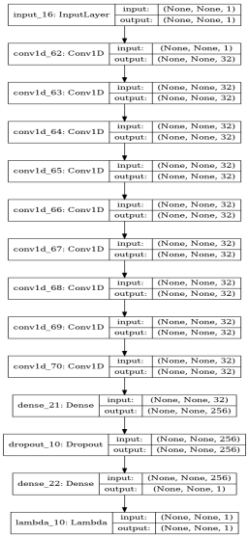
Se observa que esta red logra capturar los patrones y ciclos que se pueden tener en los diferentes productos al momento de predecir la demanda, el ajuste es muy bueno comparandolo con la data de validacion.

Posteriormente se realiza la prediccion para el paso de 1 dia y 7 dias futuros es decir 25 de septiembre, un ejemplo con las predicciones para el banano criollo.



Grafica 18. Serie Banano Criollo y predicción día.

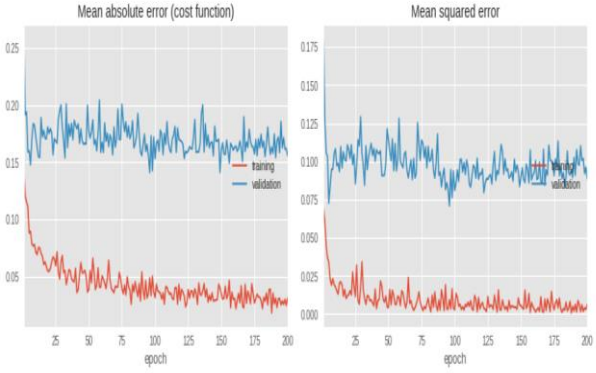
La arquitectura del modelo propuesto para predicción del precio se presenta a continuación:



Luego de un proceso de calibración de hiperparametros se usó 9 capas convolucionales para extraer patrones aproximadamente de 9 meses atrás, 32 filtros de ancho 2 por capa, tasa de dilatación exponencial (1, 2, 4, 8, ..., 256), Dropout de 0.2 y funciones de activación Relu y Linear para la capa de salida, se usó el optimizador nadam (momentum Nesterov), con learning rate de 0.001 y función de perdida MAE (Mean Absolute Error) y métrica de evaluación MSE (Mean Square Error) dado que nos permite hablar en las mismas unidades de las variables a predecir, los parámetros a entrenar fueron de 25.441 lo cual indica la efectividad de este tipo de modelos con bajo costo computacional pero gran capacidad predictiva.

Entrenamiento red neuronal para precio:

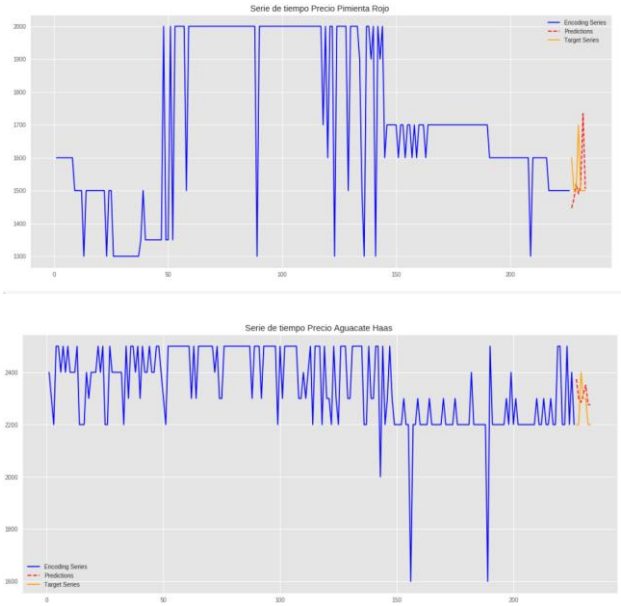
Numero de épocas 200, Batch Size 1



Grafica 19. Función de costo y error.

Se observa que el error promedio es de 0.07 y 0.08, además el entrenamiento tiene una curva exponencial decreciente lo cual da sentido al proceso de aprendizaje.

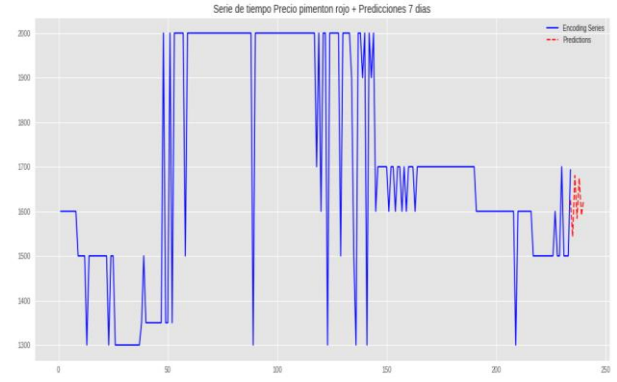
Los resultados fueron satisfactorios para visualizar el desempeño de la red se muestra un ejemplo para el caso de la pimienta roja y aguacate Haas.



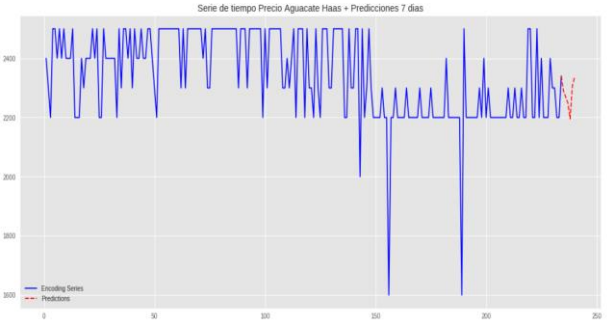
Grafica 20. Series Pimienta Roja y Aguacate y predicción día.

Se observa que el modelo recoje los comportamientos anteriores de la serie para pronosticar el precio logrando capturar la tendencia de la base de validacion.

Posteriormente se realiza la prediccion para el paso de 1 dia y 7 dias futuros es decir 25 de septiembre.



Grafica 21. Serie Pimienta y predicción 7 días.

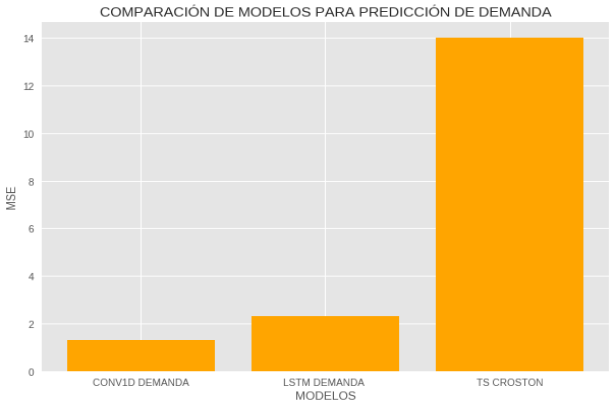


Grafica 21. Serie Aguacate y predicción 7 días.

Los modelos ANN quedaron pre-entrenados y guardados en formatos h5 para posterior uso en predicciones futuras e implementación en el sector real, los cuales se pueden consultar en el siguiente link[4], ya entrenados, es importante resaltar que el modelo funciona tanto para los 27 productos Pareto como para los 121 o más, el paso de predicción es configurable modificando una variable del modelo de acuerdo a la necesidad de la compañía.

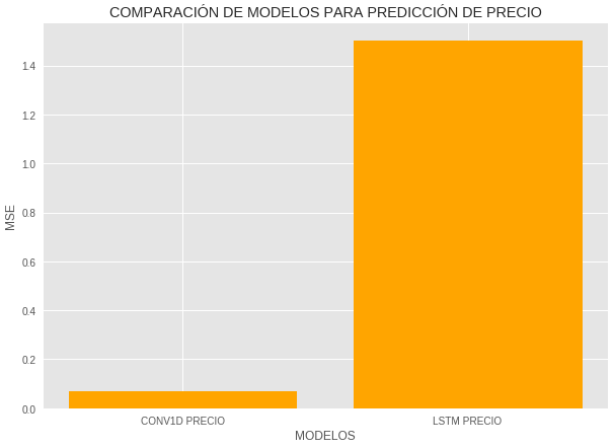
4. RESULTADOS

Los resultados se han venido mostrando de manera parcial, pero en esta sección se desean comparar los resultados de los diferentes modelos de acuerdo con la métrica de desempeño elegida (MSE).



Grafica 22. MSE de los modelos aplicados a la predicción de Demanda.

[4] https://github.com/castellwhite/Deep-Learning-Exercises-2/tree/master/Practica%201%20-%20Series%20de%20tiempo/Modelos%20Pre-



Grafica 23. MSE de los modelos aplicados a la predicción de Precio.

Se observa en las graficas anteriores que la redes convolucionales causales tiene un mejor desempeño en ambos tipos de variables a predecir, en comparacion a los otros modelos. Para el caso de los modelos CROSTON, la mayor dificultad es que el modelaje se debia realizar producto a producto lo cual generaba un alto desgaste computacional dado que el mayor interes es predecir precio y demanda de manera simultánea para cada producto, por esta razón, como modelo final se da la selección a este enfoque convolucional con resultados satisfactorios.

5. CONCLUSIONES

1. El reto analítico desarrollado en el presente documento dilucidó el poder predictivo de métodos mucho más recientes y de caja negra, teniendo en cuenta las limitaciones en términos de la información con la que se contaba y sus particularidades asociadas al tipo de sector del cual provienen, en contraste con los métodos clásicos que cuentan con un número de supuestos, que en algunos casos no se cumplen en el mundo real.

2. Los métodos de redes neuronales utilizados, a futuro podrían tener mejores resultados en su predicción para la Start-Up, siempre y cuando la base de datos presentará mayor número de registros con un espectro de tiempo mayor y con menos datos perdidos en el total de la serie.

3. Una estrategia efectiva para mejorar la previsión de productos con esta clase de datos está en la elección de aquellos que tengan la mayor importancia o mayor peso sobre las demás. Esto fue evidente en el ajuste de los modelos, donde el entrenado para los 27 productos Pareto obtuvo mejores resultados.

4. Usar la serie de tiempo completa, es decir incluir los días faltantes a las series intermitentes, no siempre brinda resultados satisfactorios, ya que, por ejemplo, para el caso de la demanda, agregó sesgo a los resultados. Al probar este enfoque se evidenció que las predicciones son lineales.

Referencias

Britz Goldie & Luong, D. (2017). *Seq2seq*. Obtenido de GitHub.

Croston, J. D. (1972). Forecasting and stock control for intermitent demands. *Journal of the operational research society*, 289-303.

Ho, T. K. (1995). Random decision forests. *Proceeding ICDAR '95 Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, 278.

Olah, C. (27 de August de 2015). *Colah's blog*. Obtenido de https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Santa Cruz R., R., & Correa, C. (2017). Previsión de demanda intermitente con métodos de series de tiempo y redes neuronales artificiales: Estudio de caso. *DYNA*.

Schmidhuber, H. y. (1997). *LONG SHORT-TERM MEMORY.* Obtenido de

https://www.bioinf.jku.at/publications/older/2604.pdf

Van den Oord, A. (08 de Septiembre de 2016). *WaveNet: A Generative Model for Raw Audio.* Obtenido de https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Referencias de internet:

[1] https://arxiv.org/pdf/1609.03499.pdf

[2] https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/

[3] https://github.com/Arturus/kaggle-web-traffic/blob/master/how_it_works.md

[4] https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html