UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE CIENCIA DE LA COMPUTACIÓN

Predicción de Precios de Cultivos Utilizando *Deep Learning*

AUTOR

Alejandro Goicochea
alejandro.goicochea@utec.edu.pe

ASESORA

Yamilet Serrano
yserrano@utec.edu.pe

Lima - Perú
2021

Resumen

La predicción de precios de cultivos es una tarea muy difícil de hacer a través del razonamiento, por ello, se suelen utilizar diferentes tipos de modelos para realizar esta tarea. Estas predicciones son útiles para decidir que cultivo se va a plantar ya que se puede saber cuál traerá mejores márgenes de ganancia. En esta investigación se detalla porque son necesarias estas herramientas predictivas para apoyar la toma de decisiones y se pretende, a través de una revisión de trabajos recientes, determinar cuál es el modelo que mejor se adapta a este tipo de problemas. Para lograr esto, se revisaron investigaciones donde comparan varios modelos de diferentes tipos y observamos cuales obtenían los mejores resultados. Al finalizar la revisión de varias investigaciones se observó que el mejor modelo para el análisis de series de tiempo es *Long Short Term Memory* y que descomponer los datos utilizando *Seasonal-Trend Decomposition Procedure Based on Loess* para utilizarlos como input también mejora los resultados.

Índice general

Capítulo 1

Contexto y Motivación

1.1. Introducción

Las series de tiempo tienen diversas aplicaciones en el área de ciencia de datos debido a la cantidad de información que se puede derivar de ellas. A veces, estos datos son estacionales como en el caso de la agricultura donde el precio de cultivos fluctúa a lo largo del año. En el área de predicción económica, las series de tiempo son de suma importancia debido a que son útiles para el análisis de regresión. Tradicionalmente, para este tipo de predicciones, estas series eran analizadas con métodos estadísticos pero con el desarrollo de nuevos modelos de *deep learning (DL)* estos se han vuelto un prominente foco de investigación.

Consecuentemente, en una revisión de la literatura reciente de esta área se encontró que este nuevo enfoque está respaldado por resultados bastante favorables. Por ejemplo, Nassar *et al.* (Nassar, Okwuchi, Saad, Karray, y Ponnambalam, 2020) compararon modelos de *machine learning (ML)*, aprendizaje profundo y estadísticos. Como resultado encontraron una mayor precisión en los modelos de aprendizaje profundo específicamente en el modelo de *Long Short-Term Memory (LSTM)* ya que este modelo fue desarrollado para tratar con el *vanishing gradient problem.* Adicionalmente, Yin *et al.* (Yin y cols., 2020) y Cao *et al.* (Cao, Li, y Li, 2019) experimentaron con modelos híbridos descomponiendo los datos brutos con diferentes métodos para usarlos como entrada ya que son datos estacionales. En ambos casos se apreció una mejora en la precisión de los modelos mostrando resultados aún más favorables cuando se hicieron modificaciones en las redes neuronales como el *Attention-based Long Short-Term Memory (ATTLSTM)* o *Convolutional Long Short-Term Memory (CNN-LSTM).* Vale notar además que en el segundo estudio se utilizaron datos del clima, datos de importaciones y de exportaciones además del precio de los cultivos para mejorar la predicción del modelo algo que no era posible en el primero debido a la disponibilidad y calidad de los datos del contexto en el que se realizó.

Todos los estudios anteriormente mencionados muestran el impacto que

tiene desarrollar un modelo para la predicción de precios de cultivos. Además, evidencian cómo los modelos deben adaptarse al contexto de cada país ya que la calidad y disponibilidad de datos no siempre es la misma. Por estos motivos, es de suma importancia realizar un modelo específico al Perú más aún cuando la agricultura es uno de los rubros con mayor crecimiento y que más aporta al PBI del país.

El propósito de este trabajo es hacer un análisis de los diferentes modelos usados para la predicción de precios de cultivos utilizando series de tiempo para luego poder realizar una selección del modelo que haya mostrado los resultados más favorables. El resto del trabajo está estructurado de la siguiente manera: El capítulo 1 termina dando una descripción y la justificación del problema. Luego, se detalla el marco teórico de la investigación. El siguiente capítulo está compuesto por la revisión de la literatura y finalmente se encuentran las conclusiones del trabajo realizado.

1.2. Descripción del problema

El sector agrícola en el Perú es de suma importancia ya que representa el 6 % del PBI nacional (Universidad Católica de Santa María, 2020). Además, mostró un crecimiento de 18.6 % anual en setiembre del 2021 (Banco Central de Reserva del Perú, 2021). Este sector además genera una gran cantidad de datos que lamentablemente no son utilizados para crear herramientas predictivas ya sea debido a falta de conocimiento o de recursos. Es importante tener estas herramientas que aprovechen las ultimas tecnologías para seguir impulsando el sector, además, se estaría acortando la brecha tecnológica entre los grandes agricultores, que si tienen la capacidad para desarrollar y utilizar estas herramientas, y los pequeños agricultores que solo se apoyan en su buen juicio para guiarse en este mercado lleno de incertidumbre. Para hacer esto, se planea desarrollar un modelo de *DL* capaz de predecir precios de cultivos el cual será puesto a disponibilidad a través de una página web de manera gratuita.

Para poder desarrollarla se identifican 4 razones principales que causan el problema de investigación. Si se logra desarrollar la herramienta descrita anteriormente y se obtienen buenos resultados, estaríamos atacando el problema de la infraestructura, los malos modelos y la habilidad humana. Por otro lado, los datos usualmente son recolectados por terceros como gobiernos o los mismos centros donde se hace la venta de los cultivos entonces se depende de estas entidades para contar con buena calidad de datos que además estén dispuestos a compartirlos con el público.

1.3. Justificación

La creación de la herramienta basada en *DL* es un tema netamente aplicativo. Mediante la creación de esta se busca poder acortar la brecha tecnológica entre las granjas pequeñas y las grandes compañías que si tienen los recursos para desarrollar modelos predictivos de alta calidad. Los agricultores van a poder utilizarla para mejorar su toma de decisiones ya que actualmente se basan en experiencia propia que no siempre resulta ser de confianza. Además, va a reducir la cantidad de alimentos no utilizados ya que en base a las predicciones se puede inferir la demanda de cada cultivo y de esta manera no se va a sobreproducir. Este balance que se genera entre la oferta y la demanda también puede llegar a mantener los precios más estables y así evitar sobrevaloración o infravaloración de productos al momento de la compra.

1.4. Objetivos

En esta investigación se tiene como objetivo principal:

- Desarrollar un modelo de *LSTM* para predecir precios de cultivos

De este objetivo principal nacen varios objetivos específicos que se deben cumplir como parte del proceso. Estos objetivos son:

- Recopilar los datos necesarios y almacenarlos en una base de datos relacional para que puedan ser utilizados fácilmente.
- Realizar una análisis exploratorio de los datos para su mejor comprensión.
- Obtener un error porcentual absoluto medio de 30 % o menor en nuestros resultados preliminares.

Capítulo 2

Marco Teórico

2.1. Datos de Cultivos

Los datos de precios de cultivos son el centro de nuestra investigación por ende es importante describir sus propiedades. Los precios de cultivos varían a diario, por eso se almacenan con una marca de tiempo y cuando se acumulan varios datos se crea una serie de tiempo. Estas series son secuencias de datos tomados con una separación constante de tiempo como diarias, mensuales o anuales. Además, se suele encontrar patrones dentro de estas series como lo son los patrones de tendencia, patrones cíclicos y patrones estacionales. En el caso de los precios de cultivos, estos siguen un patrón estacional que no se debe confundir con el patrón cíclico. Los ciclos en los datos ocurren cuando hay un crecimiento o decrecimiento en los valores pero estos no ocurren en una frecuencia determinada y suelen ser causados por factores económicos o políticos. En cambio, los patrones estacionales muestran el mismo comportamiento pero la causa y frecuencia es lo que difiere. La estacionalidad en los datos es causada por factores como el tiempo del año o día de la semana y la frecuencia es fija. Una manera de identificar este comportamiento es graficando los datos donde se apreciaría el incremento o decremento en la misma época de cada año. (Hyndman y Athanasopoulos, 2018)

2.2. Series de Tiempo

La manera en la que se analizan los datos descritos en la sección anterior ha cambiado a lo largo de los años conforme va incrementando la capacidad de procesamiento disponible y la creación de nuevos modelos. Tradicionalmente se utilizaban los métodos estadísticos ya que requerían menos poder computacional pero con el tiempo se paso a modelos de *ML* y luego de *DL* gracias al avance tecnológico y el incremento exponencial de datos disponibles. En esta sección detallaremos sus características y daremos una breve explicación de su funcionamiento.

2.2.1. Descomposición de Datos

La descomposición de datos es una técnica que facilita el análisis de series de tiempo estacionales, esto se hace aislando los diferentes componentes que se encuentran en las series. Un método para hacer esto se conoce como *Seasonal-Trend Decomposition Procedure Based on Loess* (*STL*) el cual aísla la serie (Y_t) en los componentes de tendencia (T_t), estacionalidad (S_t) y resto (R_t) (Cleveland, Cleveland, McRae, y Terpenning, 1990) Esta serie puede ser representada de la siguiente manera $Y_t = T_t + S_t + R_t$. Para lograr esta descomposición se utilizan dos bucles anidados, el bucle interior se encarga de calcular los componentes de tendencia y estacionalidad mientras que el bucle exterior esta encargado de calcular pesos basados en el resultado del bucle interior para suavizar datos atípicos en los componentes. La primera pasada en el bucle interior se calcula con un valor de 1 en los pesos y luego se utilizan los pesos calculados. Dentro de este bucle, en cada paso, primero se extrae la tendencia de Y_t y luego la estacionalidad. Una ves extraídos se restan del Y_t original para poder calcular el componente resto, el resultado de una descomposición es mostrado en la figura 2.1.

Como se mencionó en la sección anterior, *DL* saca las características de los datos automáticamente entonces no debería ser necesario descomponer series de tiempo para modelos de *DL* pero, como veremos en la sección 3, estudios recientes muestran que descomponer las series para usarlas como input brindan una mejora en los resultados modelo.

2.3. Conceptos previos de *ML*

Actualmente, para los problemas que involucran series de tiempo, los modelos de *ML* están mostrando mejores resultados (Chen y cols., 2021). El *ML* se define como un proceso automatizado que nos permite extraer patrones de datos (Kelleher y Mac Namee, 2015).

Para el problema de predicción de precios, se utiliza el aprendizaje supervisado para entrenar los modelos . Esto consiste en separar los datos en dos conjuntos, el conjunto de entrenamiento y el conjunto de prueba. Al entrenar el modelo con el conjunto de entrenamiento, este busca la relación entre las variables descriptivas y la variable objetivo que se quiere predecir. Usualmente se deben extraer características de los datos brutos para ser usados como entrada en el modelo. El objetivo del entrenamiento es crear una generalización para poder predecir datos dada una entrada de valores ya que el conjunto de entrenamiento solo es una pequeña fracción de todo el universo de datos (Bishop, 2016). El problema de esta investigación se le conoce como un problema de regresión ya que la salida consiste de una variables continua que se busca predecir con el menor error posible.

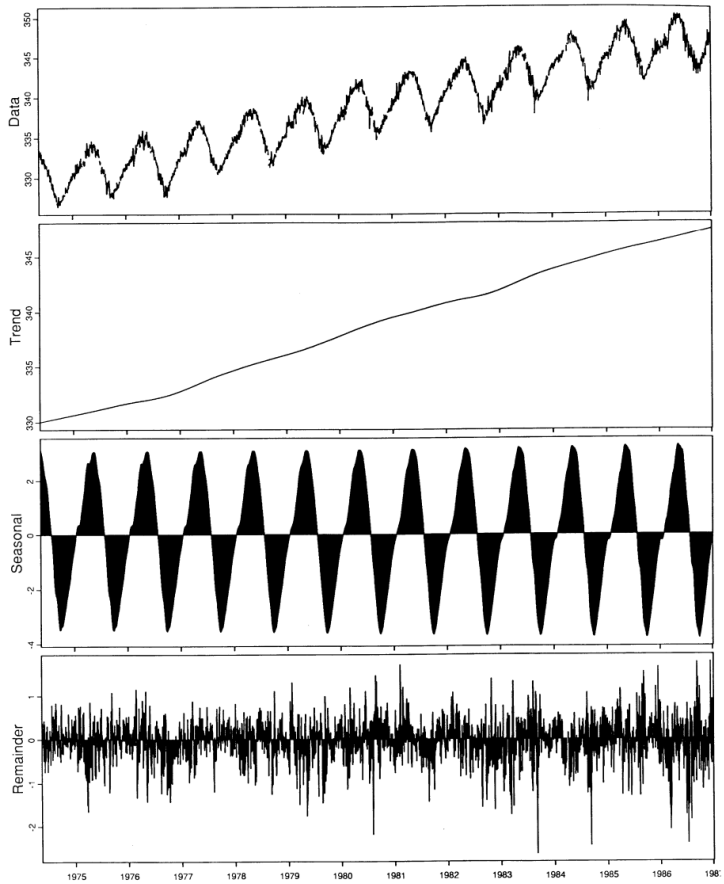

Figura 2.1: Descomposición de partículas por millón diarias de dióxido de carbono (Cleveland y cols., 1990).

2.3.1. Medición de errores

Una vez entrenado el modelo, se utiliza el conjunto de prueba para ver que tanto se acercan los valores predichos por el modelo a los esperados. En esta etapa se suele utilizar mediciones como raíz del error cuadrático medio (RECM) para cuantificar el error del modelo entrenado. Esta se calcula con la siguiente fórmula:

$$RECM = \sqrt{\frac{\sum_{i=1}^{N}(y_i - x_i)^2}{N}}, \tag{2.1}$$

donde N es la cantidad total de valores, y_i es el valor en la posición i predicho y x_i es el valor en la posición i esperado. Otras formas de medir el error del modelo son *F1 Score* y error porcentual absoluto medio (EPAM) que se calcula con la formula 2.2:

$$EPAM = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right|, \tag{2.2}$$

donde N representa la cantidad total de valores, A_i el valor esperado en la posición i y F_i el valor predicho en esta misma posición. En este estudio nos concentraremos en el EPAM.

2.3.2. *Over-fitting* y *Under-fitting*

Al entrenar los modelos hay varias consideraciones que debemos tener en cuenta para poder predecir valores minimizando la RECM. Comúnmente cuando se almacenan datos, estos vienen con un nivel de ruido que los desplaza del valor esperado. Este ruido puede ser aleatorio (p. ej. errores aleatorios de medición en sensores) pero típicamente vienen de variables que afectan los datos que no son observadas (Bishop, 2016). Cuando se trata de ajustar el modelo a este ruido para predecir los valores del conjunto de entrenamiento perfectamente se produce un *over-fitting* en el modelo. Esto va a causar que el error de las predicciones en el conjunto de entrenamiento sea mínimo o nulo pero al pasar al conjunto de prueba el error sea mucho mayor. En la figura 2.2 podemos ver una serie de puntos (azul) de una curva senoidal (verde) con un nivel de ruido añadido. Este modelo presenta un gran nivel de *over-fitting* como se aprecia en la curva roja ya que intenta se ajustar el modelo al ruido de los datos para que pase exactamente por los puntos azules. Estos puntos se pueden ver como parte del conjunto de entrenamiento y es evidente que no generalizan la curva de la cual provienen los puntos.

Por otro lado, existe el *under-fitting*, que también va a prevenir crear una generalización con conjunto de entrenamiento. La diferencia en este caso es que además de mostrar un nivel de error muy elevado en el conjunto de prueba, también lo muestra en el conjunto de entrenamiento. El *under-fitting* es mucho

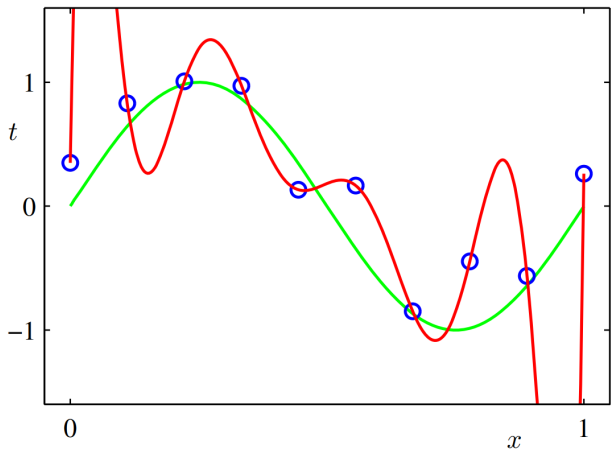

Figura 2.2: Diagrama mostrando *over-fitting* sobre puntos con ruido (Bishop, 2016).

mas sencillo de detectar ya que no hace falta probar el modelo, al entrenarlo ya muestra malos resultados. Utilizar muy pocos datos puede causar *under-fitting* ya que no se captura la verdadera tendencia o se puede estar usando un modelo lineal para generalizar una tendencia no lineal, en la figura 2.3 se muestra el segundo caso utilizando las mismas cualidades de la figura 2.2.

Cabe resaltar que al entrenar modelos de *DL*, el proceso de entrenamiento se repite varias veces, cada repetición se le conoce como una época. Si el modelo se entrena con muy pocas épocas el modelo puede presentar *under-fitting*, en cambio, muchas épocas pueden hacer que se presente *over-fitting* en el modelo. Se debe encontrar el punto donde se obtiene la mejor precisión lo que se puede hacer graficando la misma después de cada época.

2.3.3. *Deep Learning*

Los modelos de *DL* son una categoría dentro de los modelos de *ML*. Este tipo de modelos contienen varias capas de redes neuronales que buscan imitar la manera en la que un cerebro procesa información llamadas capas ocultas, la figura 2.4 muestra un diagrama de esta arquitectura. Dentro de estas redes existen nodos que almacenan, reciben y envían información a otros nodos a través de conexiones que existen entre nodos de la misma o diferentes capas. Además, las conexiones tienen peso que representa la fuerza con la que están conectados los nodos. Estos nodos realizan operaciones sobre la información que reciben propagando el resultado por todas sus conexiones y calculan el error de la predicción con una función propagándolo hacia atrás. De esta manera, en cada paso, la red puede ajustar los pesos de las conexiones y

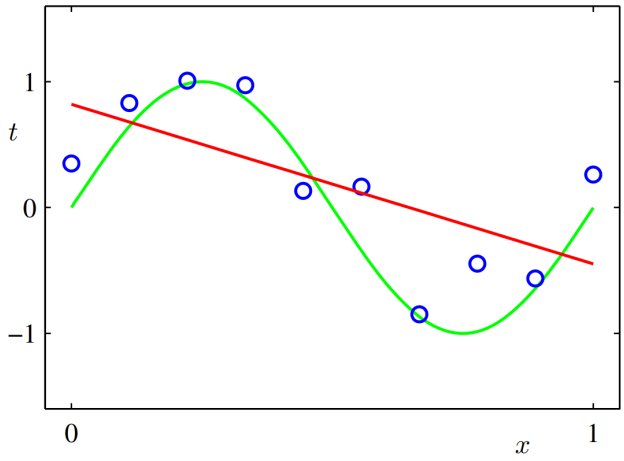

Figura 2.3: Diagrama mostrando *under-fitting* al utilizar un modelo lineal con una tendencia no lineal (Bishop, 2016).

mejorar las predicciones (Bishop, 2016). Dentro de los modelos de *DL*, existe una subcategoría llamada redes neuronales recurrentes (RNR), la mayor diferencia esta en que las predicciones no se hacen solo tomando en cuenta el input actual sino también los anteriores.

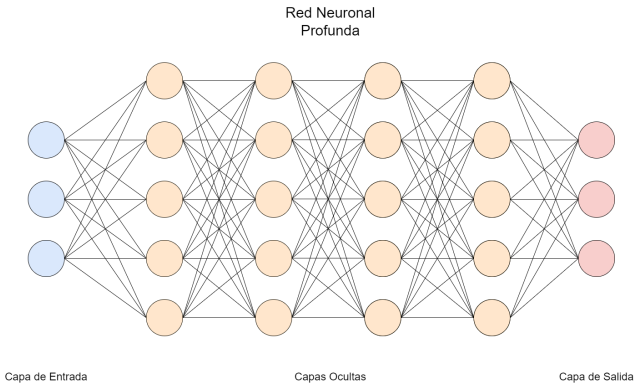

Figura 2.4: Diagrama mostrando mostrando la arquitectura de una red neuronal profunda.

La ventaja de los modelos de *DL* es que pueden aprender automáticamente mapeos arbitrarios de inputs a outputs (Brownlee, 2018). Esto es especialmente útil cuando analizamos series de tiempo ya que se trabaja con una cantidad muy grande de datos y es difícil encontrar relaciones porque

pueden ser muy sutiles o poco obvias. Otra ventaja que tienen los modelos de *DL* es que son capaces de extraer características automáticamente, reduciendo el tiempo de preprocesamiento y facilitando la creación de los modelos.

2.4. *Long Short-Term Memory (LSTM)*

LSTM (Hochreiter y Schmidhuber, 1997) es una RNR que tiene tres *gates*: *input gate, forget gate* y *output gate*. Esta estructura se muestra en la figura 2.5.

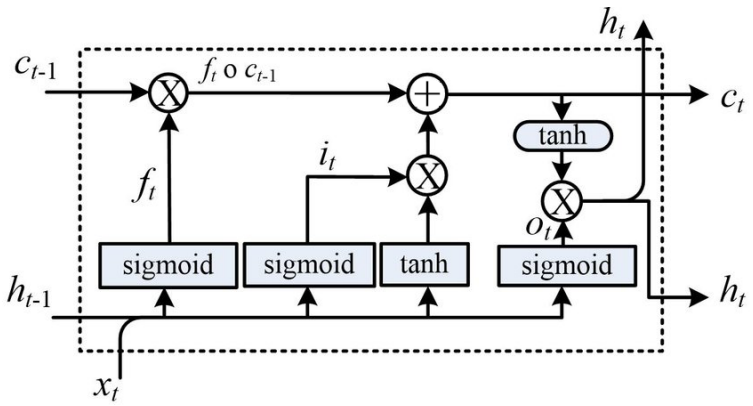

Figura 2.5: Diagrama de arquitectura en una celda *LSTM* (Jiang y Hu, 2018)

Como se observa en la figura 2.5, la celda toma como input c_{t-1} el estado de la celda o memoria a largo plazo del paso anterior $(t-1)$, h_{t-1} el estado oculto que representa el output de la celda y x_t los datos para el paso actual. Además, cada elemento de la figura representa una de las siguientes funciones:

$$f_t = sigmoid(W_f \cdot x_t + W_f \cdot h_{t-1} + b_f), \tag{2.3}$$

$$i_t = sigmoid(W_i \cdot x_t + W_i \cdot h_{t-1} + b_i), \tag{2.4}$$

$$o_t = sigmoid(W_o \cdot x_t + W_o \cdot h_{t-1} + b_o), \tag{2.5}$$

donde el parámetro W representa la matriz de pesos para cada uno de los *gates* identificado por su subíndice al igual que b el parámetro de *bias*. Los elementos circulares con una 'X' dentro simbolizan el producto de Hadamard entre dos matrices que consiste en multiplicar cada elemento en la misma posición de ambas matrices mientras que elemento con el símbolo '+' hace algo similar con la diferencia de que suma los elementos en vez de multiplicarlos.

El *forget gate* (f_t) esta a cargo de decidir que información se debe olvidar, esto se hace a través de una función sigmoide que aplasta los valores a un rango de 0 a 1 donde 0 es poco importante y 1 es muy importante. El *input gate* esta a cargo de decidir que datos del paso actual y del estado oculto anterior son

relevantes para la memoria a largo plazo, esto también se hace utilizando una función sigmoide. Finalmente, el *output gate* se encarga de actualizar el estado de la celda. Esta arquitectura fue creada para prevenir el problema de *vanishing gradient*. Este problema ocurre cuando se multiplican gradientes muy pequeñas haciendo que desaparezcan rápidamente y el modelo deje de aprender. Esto se soluciona gracias a su arquitectura ya que gracias a esta la derivada del estado no tiene un coeficiente exponencial disminuyendo la velocidad con la que la gradiente de desvanece.

Capítulo 3

Revisión Crítica de la Literatura

Una revisión reciente de la literatura muestra como la tendencia esta en comparar diferentes tipos de modelos. Nassar *et al.* (Nassar y cols., 2020) compararon tres categorías de modelos para predecir precios de cultivos: estadísticos, de *ML* y de *DL*. Se encontró que los modelos de *ML* superan a los modelos estadísticos y los modelos de *DL* superan a los modelos de *ML*. Dentro de los modelos de *DL*, los modelos compuestos superaron a los simples donde se comparo *LSTM* con *CNN-LSTM*. El aporte de la investigación es mostrar la brecha de precisión que hay entre estas categorías pero no profundiza mucho dentro de la categoría que mostró los mejores resultados.

Similarmente, Sabu *et al.* (Sabu y Kumar, 2020) compararon modelos estadísticos con modelos de *DL*. Entre los estadísticos están: *ARIMA*, *Seasonal Autoregressive Integrated Moving Average (SARIMA)* y *Holt-Winter Seasonal Method* los cuales se compararon con *LSTM*. En cuanto a *LSTM* se probó usando datos estacionarios y no estacionarios donde los datos estacionarios brindaron un resultado mucho mejor. En este caso, para predecir el precio de nueces de areca con los datos disponibles, el modelo *LSTM* obtuvo el mejor resultado con una RECM de 7.27. Esta investigación pone al modelo *LSTM* sobre los demás pero debemos notar que la cantidad de datos que se utilizaron no es suficiente para aprovecharlo al máximo al solo tener datos mensuales en un rango de 10 años.

Chen *et al.* (Chen y cols., 2021) también hicieron una comparación pero entre modelos de *ML* y *DL*. Los modelos comparados fueron: *Support Vector Regression (SVR)*, Prophet, XGBoost y *LSTM*. Igual a los estudios anteriores, este posiciona a *LSTM* como el modelo con mejor precisión. El trabajo realizado se pone disponible al público a través de una página web donde se puede elegir el periodo de predicción, una diferencia muy importante en

comparación a trabajos anteriores ya que así los agricultores que no tienen el conocimiento para desarrollar un modelo por si mismos puedan utilizarlo. Algo que se debe notar de los resultados expuestos es que el error entre los diferentes cultivos varía mucho, se podría hacer un ajuste en los parámetros del modelo o revisar la calidad de los datos para poder reducir esta brecha.

Por otro lado, como los precios de cultivos son datos estacionales, también se realizan trabajos para analizar el impacto que tiene descomponerlos y usar estos nuevos datos como input al entrenar el modelo. Méndez-Jiménez *et al.* (Méndez-Jiménez y Cárdenas-Montes, 2018) hicieron pruebas con ocho series de tiempo diferentes utilizando el método *STL* para entrenar un modelos de redes neuronales recurrentes, convolucionales y artificiales. Los resultados muestran una mejora en el rendimiento de los modelos con excepción de la red neuronal recurrente. Los autores sugieren que los datos obtenidos de la descomposición pueden ser utilizados de mejor manera y que esta puede ser la razón por la que no se obtuvo un resultado favorable en este último caso.

Un trabajo similar fue realizado por Yin *et al.* (Yin y cols., 2020) con la diferencia de que implementaron el mecanismo de atención para abordar el problema anterior. Se utilizo el mismo método de descomposición con el modelo *LSTM* y se realizaron pruebas con diferentes combinaciones de las modificaciones propuestas así como el modelo base. Los diferentes modelos probados fueron *LSTM, ATTLSTM, STL-LSTM y STL-ATTLSTM* donde el modelo *STL-ATTLSTM* tuvo los mejores resultados. A pesar de mostrar una mejora en el rendimiento a comparación del trabajo anterior, el modelo no logra manejar volatilidades muy altas como se vio en algunos cultivos.

Otro método para descomponer series de tiempo estacionales llamado *empirical mode decomposition (EMD)* es explorado por Cao *et al.* (Cao y cols., 2019). En este trabajo se exploraron modificaciones al método y probaron como estas modificaciones afectaban el rendimiento del modelo *LSTM*. Al ver que el modelo mostraba una mejora lo comparan con los modelos *SVM* y *multilayer perceptron (MLP)* pero no lograron superar los resultados alcanzados con *LSTM*. Una de las carencias de este trabajo es que solo utilizan el precio diario de los cultivos como dato para entrenar los modelos cuando podrían haber mejoras en el rendimiento si se adicionan mas parámetros financieros como el volumen del comercio.

Al culminar con la revisión de la literatura podemos apreciar como en la mayoría de los trabajos el modelo *LSTM* es el que suele tener los mejores resultados. Además, vemos como los modelos híbridos logran mejorar el rendimiento de los diferentes modelos utilizados. Específicamente para nuestro estudio, como estamos tratando con datos estacionales, era pertinente revisar como impactaba la descomposición de las series de tiempo. En esta ocasión se reviso el impacto de dos métodos, *STL* y *EMD*, donde ambos mejoraban los resultados de los modelos. Por último, notamos como varios trabajos sugerían

utilizar más parámetros además del precio diario de cultivos y la importancia que tiene la calidad de los datos que se usan.

Capítulo 4

Metodología

De manera general, en esta investigación se pretende realizar un modelo de *LSTM* con la finalidad de predecir precios de cultivos sobrepasando un límite de error. Para lograr este objetivo, se forman sub-pasos a completar los cuales son descritos en la siguiente sección

4.1. Descripción de la Metodología

La metodología se divide en cinco pasos: *Web Scraping*, Base de Datos, Análisis Exploratorio de Datos, Descomposición de Datos y finalmente Modelo *LSTM*. La figura 4.1 muestra de manera general información de cada paso y sus outputs y a continuación se describen más a fondo cada uno.

4.1.1. *Web Scraping*

Este primer paso consiste en consolidar los datos disponibles de precios de cultivos en la página del EMMSA (EMMSA, s.f.). Para hacer esto nos apoyamos de la librería de *Python* llamada *Selenium*. *Selenium* permite abrir un navegador web e interactuar con él a través de código. Esto permite automatizar todo el proceso de descarga ya que podemos introducir datos en campos de la pagina, apretar botones y extraer la información que se muestra. Dado que la información ya estaba en formato de tabla, su extracción fue sencilla y solo hizo falta escribirla a un archivo con formato de valores separados por comas (*csv*) para luego ser utilizada. En cuanto a la cantidad de información, la página web dispone de datos diarios del precio mínimo, precio promedio y precio máximo hasta el 2013. El dato más próximo se toma como el 4 de abril del 2022 ya que es el mes mas próximo con datos completos a cuando se realizó la extracción. En la figura 4.2 se ve el formato de la página. En el lado izquierdo vemos el boton de consulta, el input para la fecha de los datos y el menú donde elegimos los datos de que cultivos se quieren visualizar. En el lado derecho se ve la tabla mostrando los precios de los cultivos

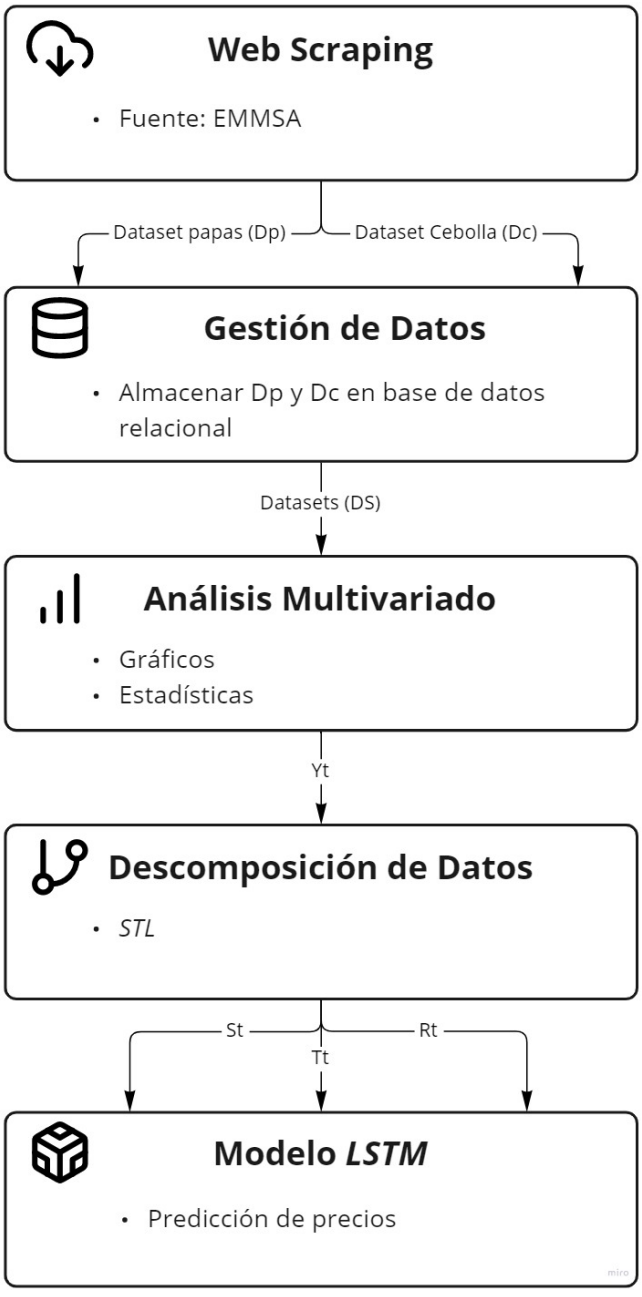

Figura 4.1: Pipeline de metodología para la investigación.

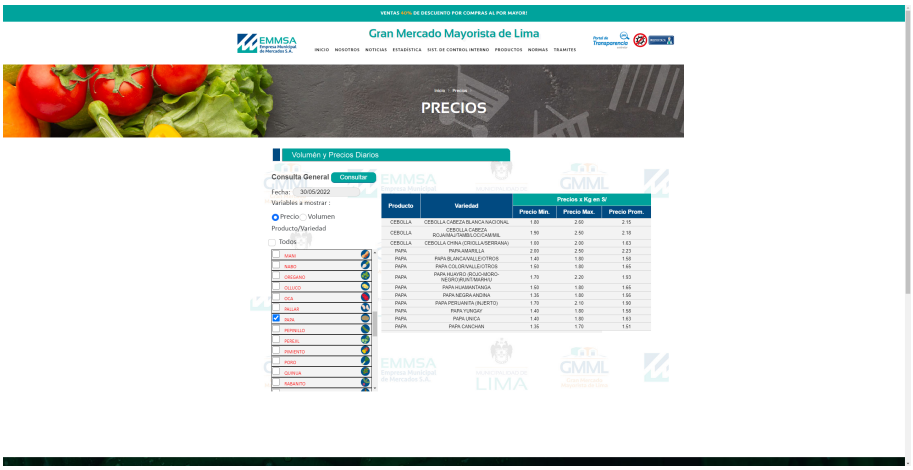

Figura 4.2: Página del EMMSA donde se realizó el *web scraping*.

seleccionados ordenados por producto, estos son los datos que se extraen de la página. El resultado de este paso fue un archivo *csv* con un dataset de papas (D_p) y un dataset de cebollas (D_c).

4.1.2. Gestión de Datos

Este paso consiste en almacenar los datos extraídos anteriormente en una base de datos relacional. Se aprovecha una base de datos ya creada anteriormente llamada AgroSmart(UTEC, s.f.) donde ya hay información de cultivos de otras fuentes almacenada. Siguiendo su formato, se almacena D_p y D_c para poder ser usada posteriormente desde cualquier dispositivo con acceso a la base de datos.

4.1.3. Análisis Multivariado

Analizar los datos en un paso sumamente importante ya que nos da una idea sobre la calidad de los datos obtenidos. Para lograrlo, nos apoyamos de gráficos y estadísticas. Algo importante que se debe resaltar es la cantidad de datos para cada tipo de cultivo dentro de los datasets. Un dataset incompleto va a dar peores resultados como se resaltó dentro del marco teórico, una de las características que necesitan las series de tiempo cuando son utilizadas para modelos de *DL* es que el período entre datos tiene que ser siempre el mismo. Los datos faltantes harían que esto no se cumpla.

El cuadro 4.1 muestra el conteo de datos para cada cultivo, podemos ver que la mayoría tiene una gran cantidad de datos. Como en el modelo se va a trabajar con promedio mensual, no es necesario descartar los cultivos que tengan pocos datos faltantes pero si es necesario descartar la papa perricholi, la papa amarilis, la papa tomasa y la papa capiro.

Cultivo	# de datos
PAPA YUNGAY	3052
CEBOLLA CHINA (CRIOLLA/SERRANA)	3052
PAPA AMARILLA	3052
PAPA CANCHAN	3052
PAPA COLOR/VALLE/OTROS	3052
PAPA HUAYRO (ROJO-MORO-NEGRO)	3052
CEBOLLA CABEZA BLANCA NACIONAL	3051
PAPA UNICA	3051
CEBOLLA CABEZA ROJA/MAJ/TAMB/LOC/CAM/MIL	3051
PAPA NEGRA ANDINA	3050
PAPA BLANCA/VALLE/OTROS	3050
PAPA PERUANITA (INJERTO)	3049
PAPA HUAMANTANGA	3029
PAPA PERRICHOLI	2238
PAPA AMARILIS	88
PAPA TOMASA	9
PAPA CAPIRO	7

Cuadro 4.1: Conteo de datos obtenidos con *web scraping*.

Luego al apoyarnos de los gráficos, podemos ver comportamientos inusuales dentro de nuestros datasets. Si alguno de los cultivos tiene una curva impredecible, lo más probable es que el modelo no tenga buenos resultados ya que los datos son muy influenciados por factores externos. Para identificar esto graficamos los datos con *boxplots* que permiten ver datos atípicos y la distribución que siguen.

La figura 4.3 muestra los *boxplots* para las diferentes cebollas de D_c donde vemos que la cebolla china tiene datos más extremos a diferencia de las otras dos que tienen una menor dispersión de datos. Esta volatilidad hace que sea difícil de predecir el precio por lo se decidió omitir este cultivo para la predicción.

Por otro lado, la figura 4.4 muestra los *boxplots* para las diferentes papas de D_p. A diferencia del gráfico anterior, aquí no hay un cultivo que tenga un nivel de volatilidad tan alto aunque si se ven más datos atípicos.

La figura 4.3 muestra los *boxplots* para las diferentes cebollas de D_c donde vemos que la cebolla china tiene datos más extremos a diferencia de las otras dos que tienen una menor dispersión de datos. Esta volatilidad hace que sea difícil de predecir el precio por lo se decidió omitir este cultivo para la predicción.

Luego de filtrar los datos que no son utilizables, calculamos el promedio mensual del precio promedio y tenemos una serie de tiempo Y_t para cada tipo de cultivo. Estas series de tiempo son mostradas en las figuras 4.5 y 4.6.

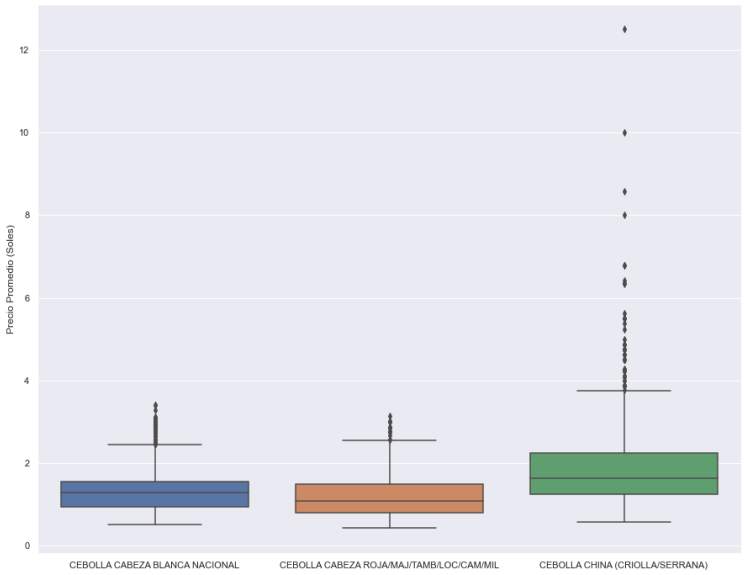

Figura 4.3: *Boxplot* mostrando precio promedio de D_c.

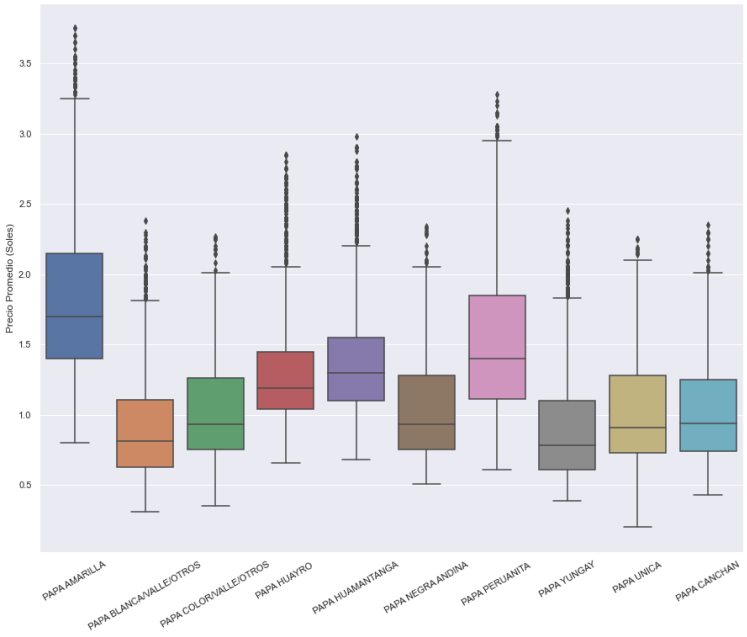

Figura 4.4: *Boxplot* mostrando precio promedio de D_p.

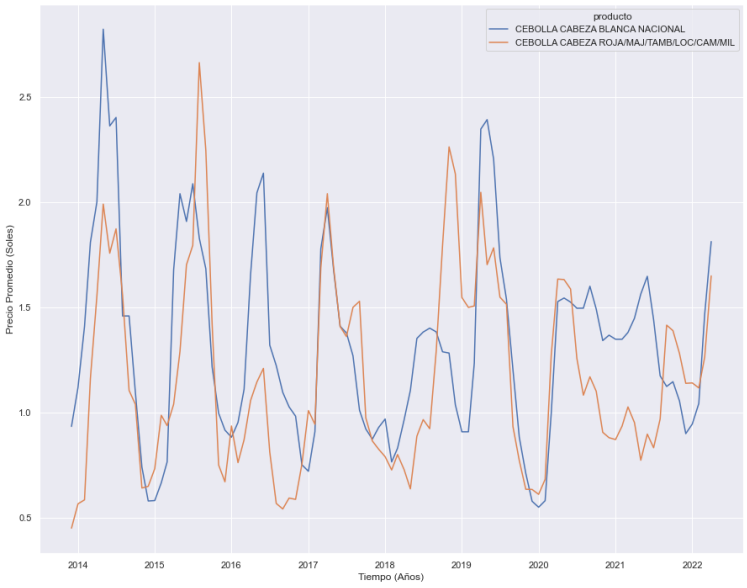

Figura 4.5: *Boxplot* mostrando precio promedio de D_c.

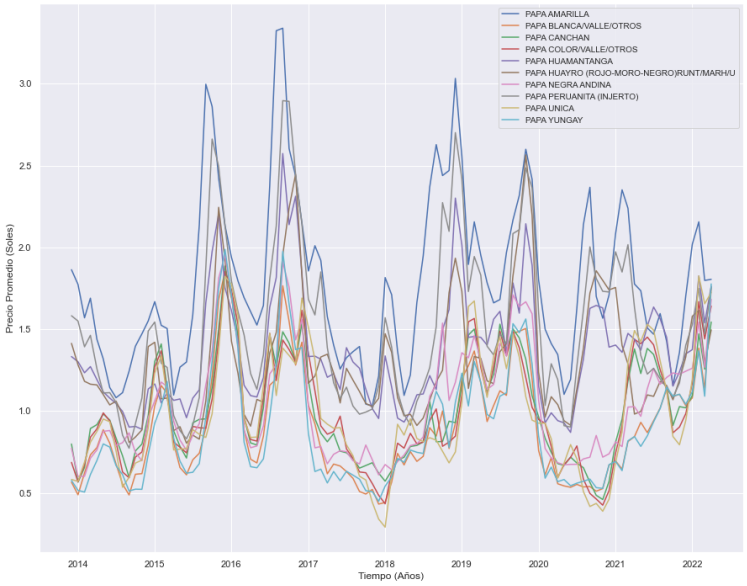

Figura 4.6: *Boxplot* mostrando precio promedio de D_p.

4.1.4. Descomposición de Datos

Una vez seleccionados los cultivos con datos limpios pasamos a la etapa de descomposición de datos usando el algoritmo *STL*. Esto se hace de manera separada para cada tipo de cultivo para calcular una serie de tiempo de la tendencia (T_t), la estacionalidad (S_t) y el resto (R_t). Estas series de tiempo son las que finalmente se usan como entrada para nuestro modelo.

4.1.5. Modelo *Long Short-Term Memory*

Este último paso consiste en correr el modelo con los datos ya preparados. Cada tipo de cultivo se tiene que correr independientemente para obtener su predicción de precios específica y cada una de las tres series de tiempo también se tiene que correr independientemente. El resultado de las series T_t, S_t y R_T se suman para obtener la predicción final con la que se calcula el EMAP que no debe sobrepasar el límite establecido en la sección de objetivos.

4.2. Alcances y Limitaciones

Como alcance en nuestra investigación tenemos solo utilizar un modelo para la predicción de precios. Además, no se harán predicciones con todos los cultivos disponibles solo con papas y cebollas ya que son los que más datos tienen.

Por otro lado, la cantidad de meses de datos en nuestro dataset se mantiene con un máximo de 100 meses debido a la capacidad computacional con la que se cuenta. Por el mismo motivo, se trabaja con una cantidad reducida de cultivos a la disponible.

Referencias

Banco Central de Reserva del Perú, B. (2021, Oct). *Producto bruto interno y demanda interna (variaciones porcentuales anualizadas) - agropecuario - agrícola.* Banco Central de Reserva del Perú. Descargado de `https://estadisticas.bcrp.gob.pe/estadisticas/series/mensuales/resultados/PN01714AM/html`

Bishop, C. M. (2016). *Pattern recognition and machine learning.* Springer.

Brownlee, J. (2018). *Deep learning for time series forecasting* (First Edition ed.). Machine Learning Mastery.

Cao, J., Li, Z., y Li, J. (2019, apr). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*, *519*, 127–139. doi: 10.1016/j.physa.2018.11.061

Chen, Z., Goh, H. S., Sin, K. L., Lim, K., Chung, N. K. H., y Liew, X. Y. (2021, jun). Automated Agriculture Commodity Price Prediction System with Machine Learning Techniques. *Advances in Science, Technology and Engineering Systems Journal*, *6*(4), 376–384. Descargado de `https://arxiv.org/abs/2106.12747v1` doi: 10.25046/aj060442

Cleveland, R. B., Cleveland, W. S., McRae, J. E., y Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion). *Journal of Official Statistics*, *6*(1), 3–73. Descargado de `http://cs.wellesley.edu/$\sim$cs315/Papers/stlstatisticalmodel.pdf`

EMMSA. (s.f.). *Volumen y precios diarios.* Descargado de `http://old.emmsa.com.pe/emmsa_spv/rpEstadistica/rptVolPreciosDiarios.php`

Hochreiter, S., y Schmidhuber, J. (1997, nov). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hyndman, R. J., y Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2nd edition ed.). Otexts, online, open-access textbook.

Jiang, L., y Hu, G. (2018, dec). Day-Ahead Price Forecasting for Electricity Market using Long-Short Term Memory Recurrent Neural Network. En *2018 15th international conference on control, automation, robotics and vision, icarcv 2018* (pp. 949–954). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICARCV.2018.8581235

Kelleher, J. D., y Mac Namee, B. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* The MIT Press.

Méndez-Jiménez, I., y Cárdenas-Montes, M. (2018). Time series decomposition for improving the forecasting performance of convolutional neural networks. En *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11160 LNAI, pp. 87–97). doi: 10.1007/978-3-030-00374-6_9

Nassar, L., Okwuchi, I. E., Saad, M., Karray, F., y Ponnambalam, K. (2020, jul). Deep Learning Based Approach for Fresh Produce Market Price Prediction. En *Proceedings of the international joint conference on neural networks.* Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/IJCNN48605.2020.9207537

Sabu, K. M., y Kumar, T. K. (2020). Predictive analytics in Agriculture: Forecasting prices of Arecanuts in Kerala. En *Procedia computer science* (Vol. 171, pp. 699–708). doi: 10.1016/j.procs.2020.04.076

Universidad Católica de Santa María, U. (2020, Jun). *El 6 % del pbi del perú lo aporta el sector agrario pese a estar relegado por el estado.* Descargado de `https://www.ucsm.edu.pe/el-6-del-pbi-del-peru-lo-aporta-el-sector-agrario-pese-estar-relegado-por-el-estado/`

UTEC. (s.f.). *Agrosmart - visualizador de precios históricos de cultivos.* Descargado de `https://compsust.utec.edu.pe/agrosmart/home`

Yin, H., Jin, D., Gu, Y. H., Park, C. J., Han, S. K., y Yoo, S. J. (2020, dec). STL-ATTLSTM: Vegetable Price Forecasting Using STL and Attention Mechanism-Based LSTM. *Agriculture 2020, Vol. 10, Page 612*, *10*(12), 612. Descargado de `https://www.mdpi.com/2077-0472/10/12/612/htmhttps://www.mdpi.com/2077-0472/10/12/612` doi: 10.3390/AGRICULTURE10120612