

Prediction Oil Price for better planification: A Machine Learning Approach

Marcelo Ortíz, Ángel Encalada

Escuela Superior Politécnica del Litoral, ESPOL

Facultad de Ciencias Sociales y Humanísticas

Facultad de Ingeniería Mecánica y Ciencias de la Producción

Campus Gustavo Galindo Km 30.5 Vía Perimetral, Guayaquil, Ecuador

Email: {maranort,angaenca}@espol.edu.ec

Abstract—Este documento realiza una comparativa del poder de predicción de algoritmos de Machine Learning versus un modelo estadístico clásico para la predicción de series de tiempo. Utilizando la serie diaria del precio de barril de petróleo WTI desde 2002 hasta 2019, se estimó modelos de regresión lineal, k-vecinos más cercanos, Autorregresivos Integrados de Media Móvil (ARIMA) y red neuronal Long Short-Term Memory con el objetivo de alcanzar una proyección fiable para la planificación en el sector público y privado. Este último algoritmo obtuvo el mejor resultado de predicción mediante la métrica de Error Cuadrático Medio.

Index Terms—Time Series / Forecast / Arima / KNN / LSTM

I. INTRODUCCIÓN

El petróleo representa uno de los principales rubros dentro de las exportaciones en la economía ecuatoriana. Por esto, el precio del petróleo es trascendental para elaboración del presupuesto general del estado. Según la Proforma Presupuestaria del Gobierno de Ecuador para el 2019, se espera tener ingresos totales por USD \$22.361 millones. Esto se proyecta tomando en consideración un precio del barril de petróleo de USD \$58,29. Además se espera que la producción petrolera se incremente en un 9%, representando una producción de 206,21 millones de barriles al año.

La correcta proyección del precio del barril de petróleo le permite al estado planificar sus ingresos y gastos. Si la proyección está sobreestimada, esto se traducirá en que el estado gastará recursos en función de un ingreso menor a lo planificado por lo que incurrirá en un déficit. Por otro lado, si la proyección está subestimada causaría que se asignen menos recursos a sectores específicos de la economía.

Como es lógico pensar, tener una idea del comportamiento futuro de los precios del barril de petróleo es vital para planificación no solo a nivel de gobierno sino también para el sector privado. Por tanto, el presente trabajo tiene como objetivo proponer 2 modelos de aprendizaje automático que sirvan de comparativa con modelos econométricos clásicos para la elaboración de un pronóstico de la serie de precios del barril West Texas Intermediate (WTI). Se espera que estos modelos sirvan como herramientas para la proyección del precio del barril de petróleo WTI y sean un insumo para la correcta planificación del sector público y privado.

II. LITERATURA PREVIA

Predecir series de tiempo en estos días puede que ya no signifique una ardua tarea, más cuando se tiene a disposición diversos algoritmos de aprendizaje de máquinas que realizan esta tarea de forma eficiente. No obstante, esto ocurre siempre y cuando las series de tiempo cumplan con las condiciones indispensables para ser tratadas directamente: la estacionariedad. Es bien sabido que las series financieras por lo general no cumplen esta condición.

Al menos de manera gráfica, el precio del barril de petróleo WTI es una serie no estacionaria. Sin embargo, con el pasar de los años diversos algoritmos se han implementado para predecir el precio aún con esta complicación. Sagheer[1] en su trabajo de investigación profundiza en la predicción de series de tiempo con redes neuronales, específicamente hablando de redes *Long Short – Term Memory*. Obteniendo resultados muy favorables de predicción en el caso del precio del petróleo demostró efectivamente que LSTM toma la información importante de la data histórica ingresada, la almacena y procesa para posteriormente predecir con base a información histórica que considera relevante.

Tang et. al [2] hace una contribución para el tratamiento de la series de tiempo no estacionarias. Es su trabajo propone hacer una combinación de PCA (Principal Component Analysis) con KNN, ya que con PCA pudo reducir la información redundante además de reducir la dimensionalidad de los features, un aumento causado por la extensión de las ventanas temporales de una serie de tiempo. Con el tratamiento de esta data logró aplicar KNN para la predicción y obtener casi un 78% de precisión en la predicción.

Sharifi [3] por otro lado, propuso una forma de predecir el vapor de agua precipitado usando SVM. Las Máquinas Vectores de Soporte analizan la serie de tiempo y corrigen el ruido causado por la no estacionariedad de la serie, haciendo que la predicción no sea tan compleja. Primero, logró obtener resultados de predicción con una desviación estándar muy pequeña respecto a la salida real, así mismo, el *bias* del modelo garantiza su fiabilidad para predecir cualquier entrada de datos.

Este documento pretende ahondar en el uso de algoritmos de *Machine Learning* para la predicción de series de tiempo, de

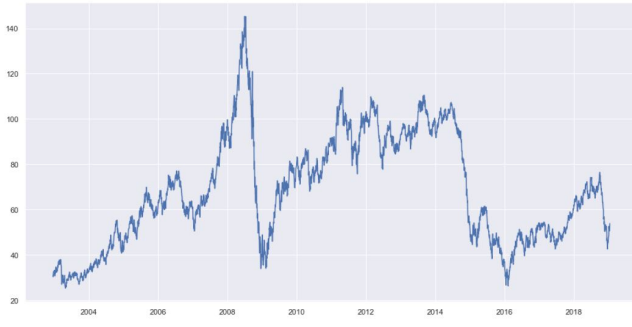


Fig. 1: Serie histórica de precios diarios Barril de Petróleo WTI

tal manera que sea una herramienta complementaria para la correcta planificación. En la siguiente sección se presente una descripción de los datos utilizados.

III. DATASET

Los datos de la serie histórica del precio del barril de petróleo fueron obtenidos de Investing.com, la cual es una plataforma financiera que ofrece cotizaciones de activos financieros e información en tiempo real de las principales bolsas de valores del mundo.

Se cuenta con información histórica desde el 19 de diciembre de 2002 hasta el 18 de enero de 2019, dando un total de 4105 observaciones.

En la figura 1 se presenta la serie de los precios del barril de petróleo que se tratará de modelar en este documento. Note que la serie exhibe una estructura no estacionaria. Además, observe que durante el año 2008, el precio subió a niveles nunca antes vistos, sobrepasando los USD \$140, siguiendo de una caída estrepitosa durante el siguiente año a un precio cercano a los USD \$35. Esto fenómeno sin duda puede ser explicado por la Gran Crisis Financiera de las hipotecas subprime del año 2008 y la posterior caída de Lehman Brothers, provocando un sentimiento de incertidumbre general en los mercados mundiales.

Otro comportamiento poco inusual de la serie se da durante el 2015 y 2016. Hubo una caída del 72% del precio del barril con respecto al 2014, donde el precio promedio fue de aproximadamente USD \$110. Durante estos años, el precio cayó por debajo de los USD \$30 debido, probablemente, a la preocupación mundial por la desaceleración de la economía china y la desafiante producción de petróleo de esquisto estadounidense. Por tanto, con la incertidumbre de una menor demanda (si la economía de China decrece, necesitará menos petróleo) y una mayor oferta, los precios se desplomaron a niveles similares al 2003.

Finalmente, para los objetivos de este artículo, se usará la información desde 2002 hasta 2017 como *training set* y la información restante como *validation set*.

IV. METODOLOGÍA

En esta sección se presenta una breve explicación de los 4 modelos utilizados para modelar la serie del precio del barril

de petróleo WTI. Para elegir qué modelo es el que realiza un mejor trabajo al momento de predecir, se usará como métrica de comparación al *Root Mean Square Error*, el cual viene dado por:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

A. Linear Regression

El primer modelo y el más básico es la regresión lineal. Este modelo permite ver la relación entre un set de variables independientes contra una variable dependiente continua. Dada la naturaleza del data set, se propuso varios *features* relacionados con la fecha de negociación que sirvan como variables predictoras del precio del barril. Estas variables pueden ser *dummies* que indiquen el año, mes, semana, día de negociación, si es inicio/fin de mes, si es inicio/fin de semestre o si es inicio/fin de año. La creación de estas variables se la realizó con la librería *fastai* en python. Por tanto, el modelo viene especificado con la siguiente estructura:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \dots + \beta_k x_{ki} + \epsilon_i \quad (2)$$

B. k-Nearest Neighbour Regression

Utilizando los mismos *features* mencionados en la sección anterior, se propone utilizar una regresión k-nn. El algoritmo de k-nn utiliza una métrica de distancia que para el caso que nos compete es la distancia euclidiana:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3)$$

El valor asignado a la estimación es computado con la media de los valores de sus vecinos más cercanos. Se propuso un rango de vecinos más cercanos de $k = 2, 3, \dots, 9$, escogiendo el mejor parámetro mediante un algoritmo de *grid search*. Grid Search es un método para encontrar la mejor combinación de hiperparámetros dado un modelo. El objetivo es estimar el modelo para un set de hiperparámetros y evaluar el que tiene el mejor rendimiento utilizando *Cross - Validation*.

C. Autoregressive Integrated Moving Average

Un algoritmo clásico para realizar *forecast* de series de tiempo son los modelos ARIMA. La mayoría de organismos gubernamentales utilizan estos modelos para realizar proyecciones de las principales variables económicas, por lo cual será el modelo *benchmark*.

Para modelar una serie con un modelo ARIMA se necesita que la serie cumpla condiciones de estacionariedad. Como es notorio a simple vista en la figura 1, la serie del precio del barril de petróleo no es estacionaria. Para corregir este problema se puede proponer varias soluciones: trabajar con la serie *detrended*, considerar fenómenos estacionales, diferenciar la serie o trabajar con modelos de series no estacionarias.

Por lo general, al trabajar con la serie diferenciada es probable que se pueda corregir problemas de no estacionariedad. Los modelos ARIMA(p,i,q) pueden estimarse mediante una serie diferenciada. De hecho, la cantidad de veces que se puede diferenciar una serie viene dada por el parámetro i del

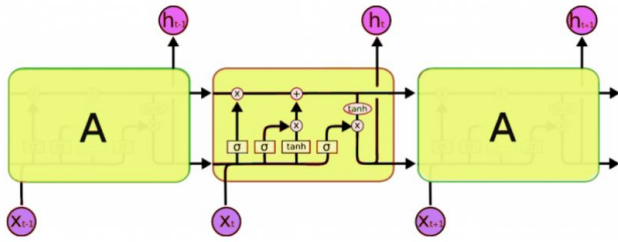


Fig. 2: Arquitectura red LSTM

modelo. Los otros parámetros, p y q son el grado de la parte autorregresiva y de media móvil, respectivamente.

Elegir estos parámetros a menudo requiere de mucho tiempo y de varios ensayos. No obstante, en Python existe una función llamada *auto_arima* que permite elegir estos parámetros de manera automática con base a los criterios de información *AIC* y *BIC*. Finalmente, la estructura del modelo *ARIMA*(p,i,q) viene dada por:

$$\Delta y_t = \sum_{i=1}^p \beta_i \Delta y_{t-i} + \sum_{i=1}^q \gamma_i \epsilon_{t-i} + \epsilon_t \quad (4)$$

D. Long Short-Term Memory Neural Network

Una red LSTM es un tipo de Red Neuronal Recurrente (*rnn*) capaz de aprender secuencias ordenadas para problemas de predicción en series de tiempo. Las aplicaciones de este tipo de redes neuronales van desde el reconocimiento del habla hasta identificación de patrones en los mercados financieros.

Bajo la premisa de que patrones del pasado pueden ocurrir nuevamente en el futuro, las redes LSTM han tomado gran popularidad en las finanzas cuantitativas debido a la habilidad de recordar u olvidar patrones selectivos de largo plazo. Por ejemplo, el precio del barril de petróleo de hoy puede depender de:

- 1) La tendencia que ha tenido la serie en días anteriores;
- 2) El precio del barril de petróleo en días anteriores;
- 3) Y otros factores externos que puedan afectar el precio el día de hoy (nuevas noticias y hechos relevantes).

Por consiguiente, es un buen enfoque tener una arquitectura capaz de discernir entre información histórica relevante y no relevante para realizar una predicción acorde.

En la figura 2 se muestra la arquitectura de una red LSTM típica. Este tipo de redes están compuestas por diferentes *memory blocks* llamadas celdas. Existen dos estados que pueden ser transferidos a las siguientes celdas: *cell state* y *hidden state*. Las celdas son las responsables de recordar información y manipularla mediante 3 mecanismos llamados *gates*: 1) forget gate; 2) input gate; y 3) output gate.

El primer *gate* es el responsable de remover la información innecesaria para la red. El segundo *gate* es responsable de la adición de información a la celda de estado. El tercer *gate* es responsable de seleccionar la información útil y mostrarla como *output*.



Fig. 3: Predicción Regresión Lineal

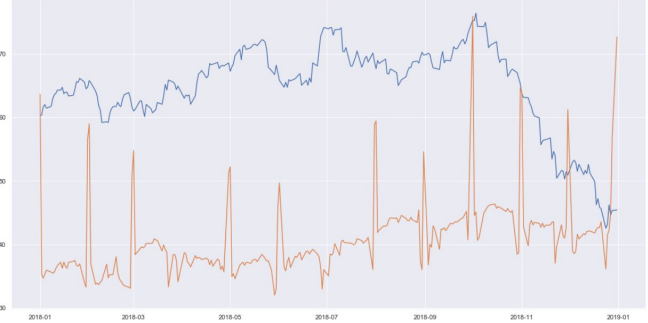


Fig. 4: Predicción K-NN, mejor $k=5$

V. RESULTADOS

En la figura 2 se graficó la serie real y la pronosticada utilizando el modelo de regresión lineal para el *validation set*. Como se puede observar, el rendimiento del algoritmo es bastante pobre y siempre tiende a sobrestimar el precio real del barril de petróleo.

Así mismo, en la figura 4 se muestra la serie real y pronosticada mediante el uso de la regresión k-nn. En este caso, k elegido utilizando *gridsearch* fue de $k = 5$. A diferencia del modelo de regresión lineal, knn tiende a subestimar el precio real del barril. Note que ambos modelos tienen un comportamiento cíclico debido a la naturaleza de los features que se están utilizando.

Por otro lado, el modelo *benchmark* presenta un mejor resultado en comparación a los dos algoritmos previos. La

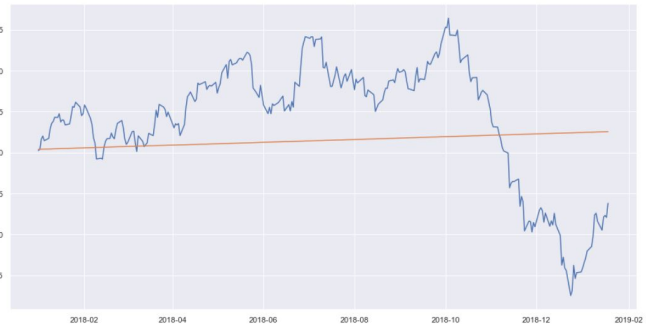


Fig. 5: Predicción ARIMA



Fig. 6: Predicción LSTM, con 50 capas

```
Fit ARIMA: order=(1, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=14056.000, BIC=14080.998, Fit time=1.200 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=14063.159, BIC=14075.658, Fit time=0.096 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=14055.707, BIC=14074.456, Fit time=0.553 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=14055.194, BIC=14073.943, Fit time=0.289 seconds
Fit ARIMA: order=(0, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=14055.157, BIC=14080.155, Fit time=1.042 seconds
Fit ARIMA: order=(1, 1, 3) seasonal_order=(0, 0, 0, 1); AIC=14055.438, BIC=14092.935, Fit time=1.661 seconds
Fit ARIMA: order=(1, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=14058.548, BIC=14089.796, Fit time=1.590 seconds
Fit ARIMA: order=(0, 1, 3) seasonal_order=(0, 0, 0, 1); AIC=14056.661, BIC=14087.909, Fit time=0.788 seconds
Total fit time: 7.261 seconds
```

Fig. 7: Resultados *auto_arima*

predicción del modelo ARIMA tiene a ser más conservadora, aunque proyecta que el precio del barril del petróleo tendrá un crecimiento paulatino. En la figura 7 se muestra un resumen de los resultados de estimación utilizando Auto Arima

En la figura 6 se muestra el resultado de la red LSTM. El resultado obtenido es bastante bueno en comparación con los otros 3 modelos. La serie proyectada tiene un comportamiento bastante similar a la serie real, por lo que da cuenta de la potencia de este tipo de algoritmos al proyectar variables financieras. Cabe recordar que LSTM se implementa por un cierto número de capas. Mientras más capas aumentamos, más exhaustivo es el algoritmo con la información que selecciona. En la Tabla 1 mostramos el error RMSE para diferente número de capas, observando claramente la disminución del error al incrementar este parámetro.

Núm. de capas	RMSE
10	4.500
20	2.373
30	1.939
40	1.615
50	1.887

TABLE I: Rendimiento de LSTM para diferente número de capas.

Finalmente, en la Tabla 2 se presenta un resumen del rendimiento en el *validation set* según la métrica propuesta. Como es notorio en el análisis gráfico, el algoritmo que presenta el mejor rendimiento es el de la red neuronal.

Algoritmo	RMSE
Linear Regression	14.3713
KNN Regression	25.8491
ARIMA	8.0410
LSTM	1.8872

TABLE II: Rendimiento de los modelos por RMSE

VI. CONCLUSIONES

En este documento se presentó un enfoque diferente para la proyección del precio del barril de petróleo WTI. Como se mencionó, la proyección de esta variable puede ser de vital importancia para la correcta planificación de un estado o cualquier agente del sector privado.

Se encontró que algoritmos de *deep learning* como LSTM pueden presentar resultados muy precisos para el *forecast* a comparación de metodologías estándar como los modelos ARIMA. Finalmente, este artículo incentiva a planificadores de entes públicos y privado a la utilización de herramientas de *machine learning* para realizar proyecciones más precisas con el objetivo de hacer eficiente la planificación presupuestaria.

REFERENCES

- [1] A. Sagheer and M. Kotb, "AC US CR," *Neurocomputing*, 2018. [Online]. Available: <https://doi.org/10.1016/j.neucom.2018.09.082>
- [2] L. Tang, H. Pan, and Y. Yao, "K-Nearest Neighbor Regression with Principal Component Analysis for Financial Time Series Prediction," pp. 127–131, 2018.
- [3] M. A. Sharifi and A. H. Souri, "A hybrid LS-HE and LS-SVM model to predict time series of precipitable water vapor derived from GPS measurements," pp. 7257–7272, 2015.