

# Mining stock price prediction using deep learning networks (LSTM Multivariable).

1<sup>st</sup> Angel Jorge Salazar

*Fac. Ciencias Económicas, Economía*  
*Universidad Nacional Mayor de San Marcos*  
Lima, Perú  
angel.jorge@unmsm.edu.pe.com

2<sup>nd</sup> Heydy Mayumy Carrasco Huaccha

*Fac. Ciencias Matemáticas, Computación Científica*  
*Universidad Nacional Mayor de San Marcos*  
Lima, Perú  
heydy.carrasco.huaccha@gmail.com

3<sup>th</sup> Jorge Sepúlveda Sepúlveda

*Facultad de Ingeniería, Ingeniería en Informática*  
*Universidad Tecnológica de Chile*  
Santiago, Chile  
buzondepitagoras@gmail.com

**Abstract**—En este artículo se presenta el análisis realizado de la data de la Compañía de minas Buenaventura en la Bolsa de valores de Lima; esta tiene como fuente la página de investing. Buscamos entrenar nuestros datos con modelo LSTM, ya que nuestra data es una serie de tiempo y además buscamos predecir los precios futuros de cierre, para una mejor toma de decisión cuando se realiza una compra o venta de las acciones de dicha compañía.

Todo el análisis fue realizado en colab con Python ya que posee muchas librerías que apoyan a este fin como por ejemplo: Pandas, Numpy, Sklearn, keras, tensorflow, matplotlib, etc. Se realizaron análisis exploratorio de cada feature, se calcularon ciertos valores estadísticos, se trazaron gráficos de acuerdo a la naturaleza de cada variable, etc.

Para este proyecto se aplicó una serie de tiempo multivariante con el modelo LSTM debido a que se busca encontrar una buena predicción del precio de cierre de la compañía minera Buenaventura en la BVL. Luego del análisis se encontró que el modelo LSTM nos arroja un loss de aproximadamente 0.19.

**Index Terms**—python, pandas, Compañía minera Buenaventura, precio de cierre, tensorflow, modelo

## I. INTRODUCCIÓN

La memoria de largo plazo (LSTM) es uno de los muchos tipos de red neuronal recurrente (RNN), también es capaz de capturar datos de etapas pasadas y usarlos para predicciones futuras. En general, una red neuronal artificial (ANN) consta de 3 capas:

- 1) Capa de entrada
- 2) Capas ocultas
- 3) Capa de salida

En una red neuronal que solo contiene una capa oculta, el número de nodos en la capa de entrada siempre depende de la dimensión de los datos, los nodos de la capa de entrada se conectan a la capa oculta a través de enlaces

llamados “sinapsis”. La relación entre cada dos nodos desde (entrada a la capa oculta), tiene un coeficiente llamado peso, que es el que toma las decisiones para las señales. El proceso de aprendizaje es naturalmente un ajuste continuo de pesos, después de completa el proceso de aprendizaje, la red neuronal artificial tendrá pesos óptimos para cada sinapsis.

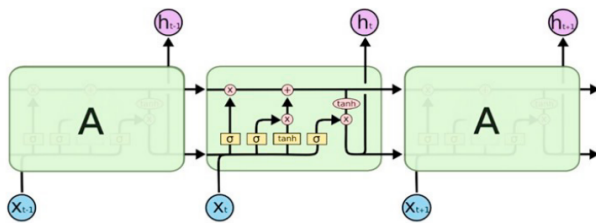
Los valores obtenidos después de esta transformación constituyen la capa de salida de nuestra red neuronal, estos valores pueden no ser la mejor salida, en este caso se aplicará un proceso de retropropagación para apuntar al valor óptimo de error, el proceso de retropropagación conecta la capa de salida a la capa oculta, enviando una señal que conforma el mejor peso con el error óptimo para el número de épocas decididas. Este proceso se repetirá intentando mejorar nuestras predicciones y minimizar el error de predicción.

Después de completar este proceso, se entrenará el modelo. Las clases de red neuronal que predicen el valor futuro en función de la secuencia de observaciones pasadas se denominan red neuronal recurrente (RNN). Este tipo de red neuronal hace uso de etapas anteriores para conocer datos y pronosticar tendencias de futuros.

Deben recordarse las primeras etapas de los datos para predecir y adivinar valores futuros, en este caso la capa oculta actúa como una reserva para la información pasada de los datos secuenciales. El término recurrente se utiliza para describir el proceso de utilizar elementos de secuencias anteriores para pronosticar datos futuros. La RNN no puede almacenar memoria a largo plazo, por lo que el uso de la memoria a corto plazo (LSTM) basada en la “línea de memoria” demuestra ser muy útil para pronosticar casos con datos a largo plazo. En un LSTM, la memorización de etapas

anteriores se puede realizar a través de puertas con una línea de memoria incorporada a lo largo. El siguiente diagrama-1 describe la composición de los nodos LSTM.

La capacidad de memorizar secuencias de datos hace que LSTM sea un tipo especial de RNN. La mayoría de los nodos de LSTM consistirán en un conjunto de celdas responsables de almacenar los flujos de datos pasados, la línea superior de cada celda vincula los modelos como línea de transporte que entrega los datos del pasado a los actuales, la independencia de las celdas ayuda al modelo a eliminar el filtro de agregar valores de una celda a otra. Al final, la capa de red neuronal sigmoidea que compone las puertas impulsa la celda a un valor óptimo al eliminar o dejar pasar los datos. Cada capa sigmoidea tiene un valor binario (0 o 1) con 0 "no deja pasar nada"; y 1 "deja que pase todo". El objetivo aquí es controlar el estado de cada celda, las puertas se controlan de la siguiente manera:



- Compuerta "olvidada": genera un número entre 0 y 1, donde 1 indica "guarda completamente esto"; mientras que 0 indica "ignorar esto por completo".
- Compuerta de memoria: elige qué datos nuevos se almacenarán en la celda. Primero, una capa sigmoidea "capa de puerta de entrada" elige qué valores se cambiarán. A continuación, una capa tanh crea un vector de nuevos valores candidatos que podrían agregarse al estado.
- Compuerta de salida: decide cuál será la salida de cada celda. El valor de salida se basará en el estado de la celda junto con los datos agregados filtrados y más recientes.

## II. TRABAJOS RELACIONADOS

Existen varios trabajos anteriores relacionados al uso de redes neuronales y del tipo Memoria de largo y corto plazo (LSTM) que son utilizados para predecir los valores de acciones por ejemplo en el rubro bursátil.

El primero tiene como título Predicción para el mercado de acciones con redes neuronales LSTM [1], en cuyo trabajo se demostró la capacidad de las redes neuronales para apoyar en la predicción del precio de las acciones. La capacidad predictiva estable y significativa de las redes neuronales utilizadas y su superioridad predictiva respecto por ejemplo a una regresión múltiple comparable, se puede ver que la memoria larga a corto plazo (LSTM) puede resolver varias tareas de series de tiempo que no pueden resolverse mediante métodos tradicionales, lo cual sugiere usar LSTM solo cuando los enfoques tradicionales más simples fallan.

El segundo tiene como título Predicción de la serie temporal del indicador bancario de referencia (IBR) con redes neuronales [2], cuyo propósito de aquel trabajo fue el demostrar la eficiencia de las redes LSTM en la generación de predicciones de series de tiempo —a través de su memoria a corto y largo plazo— que sean comparables con el modelo predictivo ARIMA para estudios econométricos.

## III. MÉTODO

La página Investing nos proporciona data de las acciones de distintas empresas; en esta caso nuestra data pertenece a la Compañía de Minas Buenaventura este pertenece al sector minero. Los datos descargados pertenecen a los años del 2013 al 2021 cuyo formato es .csv. Esta data tiene un peso de 126 KB en donde se encuentra toda la información principal a utilizar.

Ahora, este conjunto de datos consta de 1939 registros en donde encontramos los siguientes features: Fecha, Último, Apertura, Máximo, Mínimo, Vol, %var. Estos se van a analizar en el presente proyecto.

A continuación, se muestra la descripción de cada variable que nos proporciona la data:

VARIABLE	DESCRIPCIÓN
Fecha	Fecha del registro de los valores de las acciones
Último	Precio de cierre de las acciones del día
Apertura	Precio de apertura de las acciones del día
Máximo	El precio más alto de las acciones del día
Mínimo	El precio más bajo de las acciones del día
Vol.	Monto de transacción del día
% var.	Variación porcentual del precio actual respecto al precio de la transacción anterior inmediato

Se procedió a hacer la lectura de los datos en la plataforma de Google Collab, con las librerías Pandas y Numpy de Python se inició una exploración de los datos para determinar si existen vacíos en la data, cuáles son los máximos, promedios y mínimos de cada variable numérica, etc. Se hizo graficas de los datos con el uso de la librería Matplotlib y Seaborn. Y Posteriormente gracias a las librerías de keras y Tensorflow se trabajó con redes neuronales para lograr la predicción de la variable de interés

## IV. EXPERIMENTO

Se procederá a detallar cada paso realizado hasta llegar a la realización de los modelos; los tres primeros pasos pertenecen netamente al análisis de los datos.

### A. Carga de datos

Una vez que la data es cargado al drive se procede a ser llamada desde Google Colab. Luego se importa las siguientes librerías: pandas, numpy, seaborn, matplotlib, etc; una vez realizado se procede a hacer la extracción de la data. Para comprobar que se realizó satisfactoriamente, se procede a ejecutar los siguientes métodos .head() este nos muestra un vistazo de nuestra datos y .shape nos dice cuántos datos y features tiene nuestra data. Shape nos indica que se tiene 1939 registros y 7 features. Se procedió a modificar para que el feature Fecha este en formato datetime y los features Vol. y

%Var que estén en formato decimal. Adicional a este se ordenó la data de menor a mayor fecha; a continuación, se mostrará los primeros registros ordenados:

	Fecha	Último	Apertura	Máximo	Mínimo	Vol.	% var.
1938	2013-06-13	17.94	17.50	17.95	17.50	4030.0	0.0310
1937	2013-06-14	17.20	17.20	17.20	17.20	280.0	-0.0412
1936	2013-06-17	17.31	17.31	17.31	17.31	100.0	0.0064
1935	2013-06-19	17.41	17.41	17.90	17.41	700.0	0.0058
1934	2013-06-20	16.80	17.20	17.20	16.72	1770.0	-0.0350

Fig. 1. Data de las acciones de la Compañía Buenaventura

### B. Exploración de datos

Una vez cargado los datos se va a proceder a encontrar datos generales de la data: cuantos features se tiene, cuantos registros, analizar si hay datos nulos, a continuación se detalla lo encontrado.

- Se encontraron 1939 registros
- Se tiene 7 variables numéricas: 'Último', 'Apertura', 'Máximo', 'Mínimo', 'Vol.' y '% var.' etc. Y 1 variables categoricas: 'Fecha' .
- No se tiene datos nulos

### C. Elaboración de graficos

a) *Variables numéricas:* Se procede a hacer un plot para las variables numéricas; se muestra el promedio de los valor de los features al pasar los años

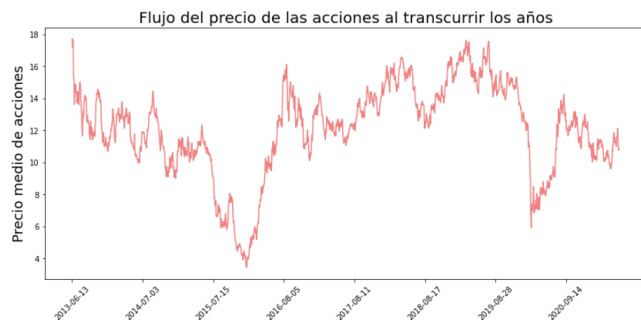
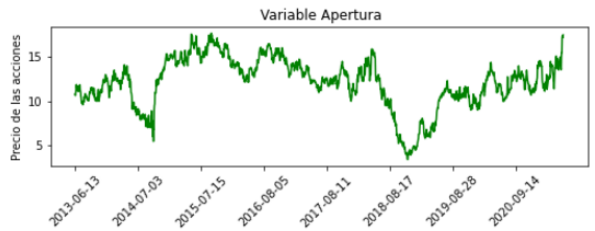
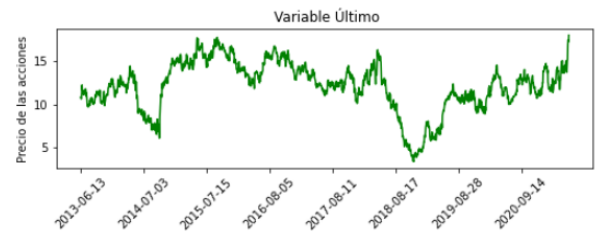


Fig. 2. Data de las acciones del 2013 al 2021

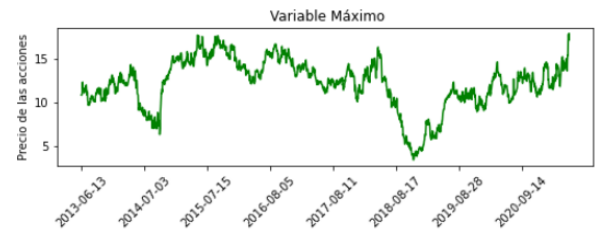
A continuación se va a observar la serie de tiempo de cada de nuestros features al transcurrir los años:

- **Variable 'Último', 'Apertura'**



Se puede observar que estos features tienden a tener un similar comportamiento, a continuación eso se observa en el grafico.

- **Variable 'Máximo', 'Mínimo'**

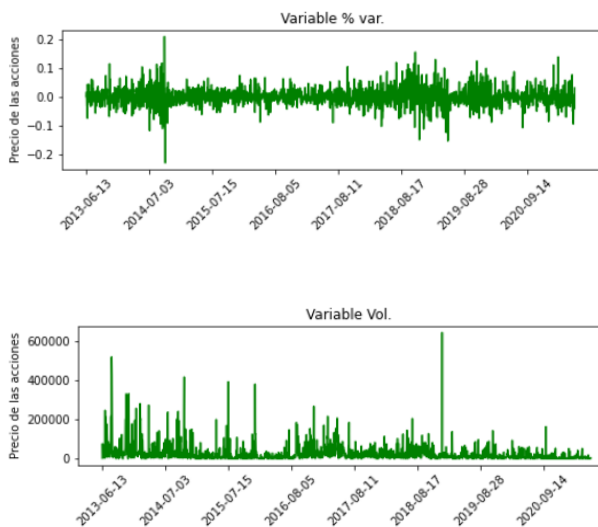


Se puede observar que estos features tienden a tener un similar comportamiento, a continuación eso se observa en el grafico..

- **Variable 'Vol', '% var'**

Se puede observar que en el feature "%var" para el año 2014 fue demasiado diversos los valores promedio de las acciones mientras que en los siguientes años se puede tener un intervalo mas cerrado de los valores entre los que fluctua los precios (variabilidad).

Y casi similar sucede en el feature de "Vol" solo que en este esos bruscos cambios se ven mas continuos.



#### D. Modelado

Para construir nuestro modelo vamos a usar series de tiempo multivariable con LSTM RNN. Para nuestro entrenamiento usamos el error cuadrático medio para optimizar nuestro modelo. Además, usamos diferentes Épocas para datos de entrenamiento (12 épocas, 25 épocas, 50 épocas y 100 épocas) nuestro modelo se estructurará de la siguiente manera:

Elegimos nuestras variables predictoras, las cuales son Apertura, Máximo, Mínimo, Vol. y var. y nuestra variable a predecir denotada es Último. Se hizo sus respectivas transformaciones a nuestras variables para ser utilizadas en nuestro series de tiempo multivariables con LSTM y finalmente nuestro dataframe queda tal como se muestra en el siguiente gráfico, queda conforme para aplicar Nuestro modelo LSTM:

Fecha	Último	Apertura	Máximo	Mínimo	Vol.	% var.
2013-06-13	17.94	17.50	17.95	17.50	4030.0	0.0310
2013-06-14	17.20	17.20	17.20	17.20	280.0	-0.0412
2013-06-17	17.31	17.31	17.31	17.31	100.0	0.0064
2013-06-19	17.41	17.41	17.90	17.41	700.0	0.0058
2013-06-20	16.80	17.20	17.20	16.72	1770.0	-0.0350

Fig. 3. Variables predictoras y a predecir.

Luego se procede a importar la librería sklearn, tensorflow, keras, etc el cual nos permitirá crear nuestra red neuronal y poder desarrollar un modelo de series temporales multivariables con LSTM y su respectiva métrica. Para esta investigación se analizará un modelo de series temporales multivariable con LSTM y poder entrenar para que nos brinde predicciones futuras de precios de cierre para tomar decisiones objetivas en la compra o venta de acciones de la compañía Buenaventura, Además se toma en cuenta la métrica del accuracy y loss(pérdida)="mean\_squared\_error".

a) *Modelo series temporales multivariables con LSTM:*  
Al desarrollar este modelo LSTM, podemos obtener estos 3 figuras importantes que nos ayudaran a entender el modelo y de que forma se trabajó y que resultados nos arrojan.

#### • Estructura de la Red neuronal

En el gráfico podemos visualizar la estructura de nuestra red. Tenemos 2 capas ocultas; En la 1ra capa oculta estamos inicializando nuestro modelo como secuencial con 64 neuronas Y para la 2da capa tenemos 10 neuronas luego de este; estamos realizando un dropout para apagar un 25% de las neuronas de la capa. Para nuestra capa de salida se uso la función de activación lineal y nos dará un unico valor de salida.

Al compilar nuestro modelo estamos usando la métrica de loss (mean squared error) y el optimizador de Adam con un learningrate=0.01.

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
=====		
lstm_2 (LSTM)	(None, 90, 64)	17920
lstm_3 (LSTM)	(None, 10)	3000
dropout_1 (Dropout)	(None, 10)	0
dense_1 (Dense)	(None, 1)	11
=====		
Total params: 20,931		
Trainable params: 20,931		
Non-trainable params: 0		

Fig. 4. Estructura de nuestra red.

#### • Métrica

En el siguiente gráfico podemos visualizar la métrica de la pérdida con las cuales podemos tomar la decisión de nuestro modelo, en nuestro trabajo buscamos que tenga el menor valor posible de pérdida cada vez se va iterando. Además nos enfocamos en esta métrica(pérdida), porque hace fácil el poder tomar la decisión de cuantas iteraciones debemos hacer en nuestro modelo. Con LSTM obtenemos un  $mean\_squared\_error$  de 0.1896.

```
Epoch 00012: val_loss did not improve from 1.07563
Epoch 13/38
45/45 [=====] - 4s 79ms/step - loss: 0.1896 - accuracy: 0.0000e+00 - val_loss: 1.3546 - val_accuracy: 0.0000e+00
Epoch 00013: ReduceLROnPlateau reducing learning rate to 0.004999999988241291.
Epoch 00013: val_loss did not improve from 1.07563
Epoch 00013: early stopping
CPU times: user 20s, sys: 3.64 s, total: 24s
Wall time: 51.5 s
```

Fig. 5. Métrica del modelo.

## • Predicción

En nuestra representación gráfica podemos comparar datos entrenados, reales y la predicción futura de los precios de cierre.

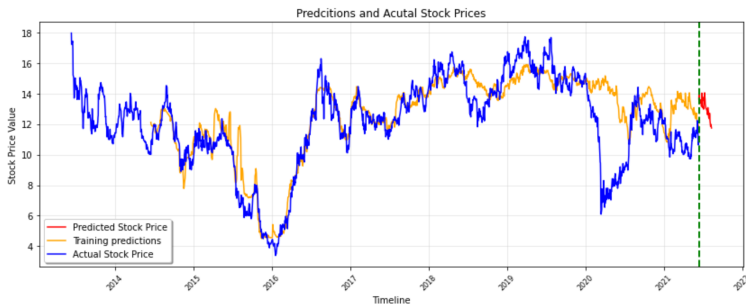


Fig. 6. Predicción.

## V. CONCLUSIONES Y RECOMENDACIONES

Luego de realizar nuestro análisis y modelo podemos afirmar lo siguiente:

- Nuestro modelo es capaz de predecir los precios de cierre de los activos, tiene baja pérdida y una precisión muy baja.
- Con nuestro modelo LSTM, se buscó predecir el precio de cierre de la compañía de minas Buenaventura en la BVL, de las cuales según el modelo trabajo, nuestra red aprendió bien, para realizar una predicción, futura aunque la precisión es muy baja.
- Se observó que nuestro precio de cierre de los activos se encuentra entre [3.39, 17.69]
- Nuestro modelo esta prediciendo el valor de cierre de los activos con el valor de 0.1896 para la metrica loss.
- Sería recomendable tener un mayor conjunto para los datos, esto conllevaria a modificar el número de épocas a fin de buscar que este se adapte a la naturaleza de nuestros activos y así poder maximizar la precisión de nuestras predicciones.
- Se tiene que mejorar el modelo en cuanto a la precisión.

## REFERENCES

- [1] D. Herrera, Predicción para el mercado de acciones con redes neuronales LSTM, Tesis Maestría Ingeniería y Analítica de Datos, Universidad Jorge Tadeo Lozano, Bogotá, Colombia, 2020.
- [2] G. Mondragón, Ó Granados, y O. García-Bedoya, Predicción de la serie temporal del indicador bancario de referencia (IBR) con redes neuronales, Revista Mutis, vol. 11, n.º 1, 2021 .

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.