

A UN TWEET DE CAMBIAR LA ECONOMIA COLOMBIANA

Proyecto Inteligencia Artificial

Es innegable el impacto que las redes sociales tienen en nuestro día a día, la mayor parte de la población las usa a diario y frecuentemente, ya sea para comunicarse con alguien a pesar de la distancia, para informarse sobre los acontecimientos y noticias, para compartir puntos de vista e incluso, entretenerse por horas con videos de Tik Tok. A nivel personal, nuestro uso de las redes sociales no tendrá un gran impacto, ya sea con algo que compartimos o algo que escribimos en ellas, es poco probable que impactemos a muchas personas con un par de frases en Twitter. ¿Y si con un tweet empeoramos la economía de todo un país?; no es descabellado pensar que en los tiempos que corren un paso en falso de un político podría cambiar las predicciones y el estado de la moneda local con relación al mercado internacional, por lo que se busca analizar si los cambios en el valor de nuestra moneda colombiana respecto al dólar esta relacionada con las acciones que lleve a cabo un jefe de estado y como estas sean comunicadas por su red social predilecta.

Objetivos

- Analizar los datos de interés para ver la relación entre los tweets del presidente y las variaciones del dólar y como estas afectan nuestra moneda local; este preprocesamiento a los datos de debe hacer principalmente con un historial del precio del dólar y un compilado de los tweets del mandatario para identificar las palabras clave que pueden causar un cambio en la economía.
- Clasificar los datos obtenidos para determinar su relevancia e impacto en las variaciones del dólar, ya que por medio de series de tiempo será posible visualizar como se comportan las estimaciones de su valor en situaciones normales y como estas se ven afectadas cuando se presenta una novedad en Twitter por parte del jefe de estado, esto teniendo como base las ANN.

Programación del Proyecto

Con la ayuda de Python fue posible realizar el analisis de los datos base del proyecto, los cuales comprenden por una parte, los valores relacionados con el dolar, su precio de apertura, valor minimo y maximo, todo esto relacionado con una fecha con la que podremos analizar el cambio del valor del dolar; todos estos datos se convirtieron a pesos colombianos para poder aterrizar estos datos a nuestro contexto. La segunda parte de estos datos corresponden a los tweets del presidente actual de Colombia, esto debido a que en los ultimos meses, nuestro país a pasado por dificultades a nivel economico y según los medios de comunicación, este aumento en el precio del dolar que produce un aumento global en los precios de las naciones puede ser causado por la incertidumbre que genera el mandatario en su red social predilecta. Por ello, se buscará analizar con series de tiempo estos datos para realizar la predicción del valor del dolar en nuestro contexto y como los tweets con sus caracteres y fecha de publicación pueden afectar a los valores.

Librerías utilizadas

A continuación, se muestran los módulos utilizados en el código de Python, con las librerías necesarias para trabajar con operaciones aritmeticas en Python, manejo de estructuras de datos y bibliotecas que encontramos en sklearn para la producción de modelos de aprendizaje supervisado y Machine Learning.

```
#Librerias para trabajo en Python con operaciones aritmeticas y datos
import pandas as pd
import numpy as np
#Modulos de sklearn, utiles para ML
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer, r2_score
#Librerias de Keras para series de tiempo con ANN
from keras.wrappers.scikit_learn import KerasRegressor
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping
from keras.optimizers import Adam
from keras.layers import LSTM
#Para graficar los datos en Python
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Pre-Procesamiento de los datos

Este apartado es importante previo a trabajar con los datos, debido a que los valores atipicos, valores nulos o elementos que no se pueden analizar con Python pueden producir errores a la hora de ejecutar los códigos y afectar los resultados que deberíamos obtener. A continuación, se muestran algunos de los procedimientos elaborados para la preparación de los datos.

```
# verificar si hay valores nulos en los datos
print(df.isnull().sum())
# Eliminar filas con valores faltantes (NaN)
df.dropna(inplace=True)
# Eliminar filas duplicadas
df.drop_duplicates(inplace=True)
# Resetear los índices del DataFrame para reorganizar los índices de las filas ya que eliminamos filas en el Dataframe
df.reset_index(drop=True, inplace=True)
# verificar si hay valores nulos en los datos
print(df.isnull().sum())
# Seleccionar las columnas numéricas que deseas normalizar
columns_to_normalize = [ 'Open', 'High', 'Close']
```

Graficas de los datos y la visualización de conjunto de entrenamiento y test

A continuación, se muestra como se realizaron las graficas con los precios historicos del dolar en pesos colombianos y como se repartieron estos datos para en entrenamiento del modelo y su testeo.

```

#Grafica con los datos de la variación del dolar en COP
ind_df = ind_df.sort_index()
plt.figure(figsize=(10, 6))
ind_df['Adj Close'].plot();

#Grafica con la partición de los datos (Entrenamiento y test)
df2 = df['Adj Close']
partition=195
train = df2.loc[:partition]
test = df2.loc[partition:]
plt.figure(figsize=(10, 6))
ax = train.plot()
test.plot(ax=ax)
plt.legend(['train', 'test']);
print(df2.shape)
print(train.shape)
print(test.shape)

```

Preparación de los datos para el modelo de series de tiempo

Con los módulos de sklearn, se realizó la preparación del conjunto de datos de entrenamiento y de prueba.

```

#Preparación de los datos de entrenamiento y testeo
scaler = MinMaxScaler()
scaler.fit(train.values.reshape(-1, 1))
train_sc = scaler.transform(train.values.reshape(-1, 1))
test_sc = scaler.transform(test.values.reshape(-1, 1))

X_train = train_sc[:-1]
y_train = train_sc[1:]
X_test = test_sc[:-1]
y_test = test_sc[1:]

```

Resultados y validaciones

Todos los resultados obtenidos y las metricas utilizadas para valorar la efesctividad del modelo, se muestran en el código fuente explicado en las etapas anteriores y que se encuentra en la carpeta del [GitHub de este proyecto](#).