www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026,132-147. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i2.1257

The effect of inflation and the economy during the Pandemic years: A Methodological Proposal for Sentiment Analysis in Python

Maria Beatriz Bernábe Loranca [1], *Melissa Mendoza Bernabe* [2], *Alberto Carrillo Canán* [1]
[1] Benemérita Universidad Autónoma de Puebla, 14 Sur y Avenida San Claudio, Puebla.
[2] Universidad Iberoamericana, Camino Real A San Andrés Cholula No. 4002, Puebla, Pue.
beatriz.bernabe@gmail.com, melrosa88@gmail.com, alberto.carrillo@correo.buap.mx

Abstract. In this work, NLP Natural Language Processing has been used, specifically sentiment analysis for the problem "The effect of inflation and the economy during the pandemic years" using as a database a tweet written between the dates 01 - 01-2020 to 10-11-2022. The objective is to classify tweets as positive, negative and neutral and establish the impact of some factors related to the economy after the Covid-19 pandemic. A prediction was made based on historical data, on how inflation could be modified in the period 2025-2028. This analysis provides a method to track public inflation expectations by tracking opinions from rich network data.
Keywords: NLP, sentiment analysis, inflation.

Article Info
Received Dec 26, 2025
Accepted Jan 26, 2026

1 Introduction

It is considered that sentiment analysis traces its origins to 2004, although the use of Natural Language Processing (NLP) techniques emerged around the year 2000. This field encompasses everything from machine learning to sentiment analysis using text categorization techniques for subjective sentences, with the purpose of analyzing large volumes of content from the Internet, primarily from social media (Pang et al., 2002). Currently, sentiment analysis is used to extract information from the Internet, such as texts, tweets, blogs, social networks, news articles, reviews, comments, and more, employing various techniques. Within NLP, statistical and machine learning methods are found, which organizations, research centers, and others successfully utilize to analyze and evaluate extracted information, identifying new opportunities and better targeting their message to their intended demographic.

With the success of social networks like Facebook and Twitter, the growing popularity of blogs, and rating or recommendation sites, the power of NLP and sentiment analysis has become even stronger for filtering and analyzing information to better understand consumer sentiment and make informed decisions. This article analyzes public opinions on post-Covid inflation. The data was obtained through a Twitter tweet download procedure. Twitter was chosen due to the diversity and variability of users generating opinions and experiences, which can be evaluated. By analyzing user tweets on a specific topic, it is possible to determine the average general sentiment on social media regarding inflation during the COVID-19 pandemic. For the entire development of this work, the Python programming language was used, along with the NLTK, RE, Pandas, NumPy, and TextBlob libraries.

The paper is structured as follows: This introduction as Section 1. A brief state of the art is presented in Section 2, leading to the architecture of the developed program, detailed in Section 3. Section 4 describes the dictionary acquisition process, followed by Section 5, which outlines the sentiment analysis procedure. Section 6 presents an inflation prediction covering the period from 2025 to 2028. Section 7 discusses the results obtained and compares them with those found in the literature. Finally, the conclusion of the results obtained in the study is presented.

2 State of the Art

Inflation can be measured in general terms, but food inflation has eroded global living standards in recent years. A similar story applies to energy costs, which are reflected in higher transportation expenses. Prices of other essential items are also rising. The relative impact of inflation on food, energy, and other priority items varies considerably between countries; however, the global

trend during and after the COVID-19 pandemic has been upward. In Mexico's case in 2022, the observed increase, according to Esquivel (2022), responds to exogenous phenomena in both core and headline inflation. Three key factors in our country are: 1) the rise in global inflation, 2) the trend of prices that grew less than usual during 2020 due to the pandemic and persist, causing temporarily higher annual inflation rates, and 3) changes in household consumption patterns during the pandemic, which have led to price and consumption adjustments.

The data show the continuous advance of inflation and the pandemic's impact on it. Muller Durán (2022) argues that the current inflation experienced by several countries, including Mexico, is a consequence of supply and demand imbalances in the goods market caused by the pandemic and primarily quarantine measures. In his study, he uses a Structural Vector Autoregressive (SVAR) model for a panel of seven industrialized and emerging countries during the 2020-2021 period. This macroeconomic perspective measures COVID's impact on inflation but does not address public opinion and perception (Muller Durán, 2022). The Bank of Mexico, in its 2020 report, presents an analysis of inflation trends and finds interesting behavioral details in certain items. For instance, it notes a general price decline in the first period of the pandemic, followed by rapid increases in fuel and food prices in most countries afterward. For this study, the change in demand due to lockdowns is notable, with inflation effects being more pronounced in developing countries (Banco de México, 2020).

Public perception and opinions about inflation vary according to educational and economic levels. This assertion is supported by qualitative research on post-COVID-19 perception in Mexico, conducted via WhatsApp surveys, which integrates the contextualization of the health and economic crisis from a historical perspective (Nava Olivares, 2020). Globally, studies on perception and opinions about inflation during COVID have been evaluated by news outlets or government institutions such as the Bank of Mexico, the World Bank, or the Federal Reserve Bank of New York (Kohli et al., 2022). The bibliographic research shares the conclusion of increased uncertainty about inflation, as well as the tendency for consumers with lower education levels to have higher inflation expectations, with pronounced inflation concerns across all socioeconomic strata. The opinion studies found focus on consumer surveys, which are then subjected to statistical perception analysis. In other cases, inflation studies are economic in nature, modeled with econometrics and supported by mathematical tools. Generally, these studies do not use natural language processing or social media sentiment analysis; therefore, this work adopts this approach. Li and Tang (2022) developed a study using Twitter and Google to reveal opinions about inflation during COVID by evaluating lexical orientation for sentiment analysis. Their method includes logistic regression and random forest for sentiment classification under the BERT model. Morales Pelagio, Robles Ulloa, and Mora Gutiérrez (2024) compared inflation and unemployment in the United States and Mexico during the onset of the COVID-19 pandemic, using a Generalized Additive Model (GAM). This model is used to study the (nonlinear) relationship between inflation and unemployment in both countries. García Pérez and Mendoza Rivera (2025) modeled inflation expectations in Mexico by combining a survey of experts with a Bayesian inference model. The results suggest that the most likely scenario for the next 12 months is high inflation (8%), with a posterior probability of 62.1%. The analysis concludes that inflation could remain high due to external factors and mistrust in monetary policy.

3 Architecture: Required Programs and Downloads

Throughout this work, Python 3 programming language was used with Jupyter Notebooks, a web application that supports real-time code execution, mathematical equations, visualization, and Markdown in the Visual Studio Code editor. The implementation focused on three main programs: a) "DOWNLOAD_TWEETS.ipynb": Downloads tweets with selected keywords and preprocesses the tweets to store them in a "CSV" file, b) "TWEET_SENTIMENTS.ipynb": Classifies each tweet as positive, negative or neutral and creates dictionaries with the most frequently used words, and c) "TWEET_ANALYSIS.ipynb": Analyzes the obtained results.

3.1 Download Algorithm

To collect the tweets, the Twitter module from the Snscrape library (JustAnotherArchivist, n.d.) was utilized. This library is a common choice for its effectiveness as a social media scraping tool, capable of retrieving public data such as user profiles, hashtags, or specific search results. The process was executed using a custom-formatted query to find the relevant tweets. The structure of this query is as follows:

```
"(#hashtag_word) lang:en until:end_date since:start_date -filter:links -filter:replies"
```

Where:
```
#hashtag_word: Represents the selected keyword for obtaining tweets.
lang:en: Specifies that only tweets in English should be collected.
```

```
until:end_date since:start_date: Defines the specific date range for the tweet extraction.
-filter:links -filter:replies: Filters out tweets that contain links or are replies to other
tweets, ensuring a focus on original content.
```

The data collected was stored in CSV files. The implementation of the download algorithm is described in the following pseudocode:

```
BEGIN
    OPEN csv_file
    SET query = "(hashtag_word) lang:en until:end_date since:start_date -filter:links -
filter:replies"
    FOR EACH tweet IN TwitterSearchScraper(query).get_items()
        IF tweet_limit IS reached THEN BREAK
        cleaned_text = cleanTweet(tweet.content)
        WRITE [tweet.date, cleaned_text, tweet.hashtags] TO csv_file
    END FOR
    CLOSE csv_file
END
```

3.2 Word Selection

Following the tweet downloads, a total of 14 words related to public perception were selected for "The effect of inflation and the economy during the pandemic years". These words are connected to economy, employment, money, recession, price of goods, and inflation: #inflation, #economy, #StockMarket, #price, #salary, #rise, #poverty, #debt, #currency, #money, #devaluation, #recession, #deflation, #employment.

These words serve as identifiers to group opinions, and for each one, 3 different searches were conducted divided into important dates of the pandemic. For this problem, we considered performing 3 queries per hashtag or word, which were manually modified:

```
query1: we obtain tweets from 2020-01-01 to 2020-03-13, important dates as they mark the
beginning of the pandemic.
"(#inflation) lang:en until:2020-03-13 since:2020-01-01 -filter:links -filter:replies"
query2: we obtain tweets from 2020-03-13 to 2021-03-13, marking the first year of the pandemic.
"(#inflation) lang:en until:2021-03-13 since:2020-03-13 -filter:links -filter:replies"
query3: we obtain tweets from 2021-03-13 to 2022-11-10, marking the second year of the pandemic
and current dates.
"(#inflation) lang:en until:2022-11-10 since:2021-03-13 -filter:links -filter:replies"
```

3.3 Data Cleaning

An initial step for identifying sentiments in tweets is preprocessing. Techniques are applied to the data to reduce both text noise and dimensionality, helping to improve classification effectiveness.

For data cleaning, a function called cleanTweet(tweet) was coded, whose parameter is the text or content of the tweet. The NLTK library (Bird et al., n.d.) is essential for cleaning tweets and involves converting the tweet to lowercase, removing punctuation, emojis, eliminating stop words or empty words, and numbers, to finally lemmatize the tweet. It is known that lemmatization is the process of grouping different inflected forms of a word so they can be analyzed as a single item, and NLTK with the WordNetLemmatizer() function serves this purpose. The code associated with this process can be seen in Data Cleaning Process Code.

4 Polarity

To classify tweets as positive, negative, and neutral, TextBlob (Loria, n.d.) was used, which is a high-level library built upon the NLTK library. TextBlob has a module called sentiment that obtains the subjectivity and polarity of tweets. The polarity score is a value within the range [-1.0, 1.0]. Subjectivity is a value within the range [0.0, 1.0] where 0.0 is very objective (i.e., does not express strong sentiment) and 1.0 is very subjective. TextBlob uses a set of words that have been labeled as positive or negative, so the training data is labeled to be processed in a Naive Bayes classifier. For this purpose, a function called getPolarity(tweet)

was used where the tweet is a parameter, and the sentiment.polarity function is applied to obtain the tweet's polarity between -1 and 1.

The following two functions allow obtaining the subjectivity and polarity of tweets, which are added to the dataframe as new columns called "Subjectivity" and "Polarity":

```
def getPolarity(text):
    return TextBlob(text).sentiment.polarity
```

On the other hand, the function called getSubjectivity(tweet) receives the tweet as a parameter while sentiment.subjectivity obtains the subjectivity:

```
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity
df["Subjectivity"] = df[1].apply(getSubjectivity)
df["Polarity"] = df[1].apply(getPolarity)
```

The dataframe output is shown in Figure 1:

	1	Subjectivity	Polarity
0	u economy recession yet #cnbc #dowjones #econ...	0.200000	0.200000
1	smelling #inflation far distant future got #g...	0.491667	0.000000
2	digitally print u would considered counter...	0.000000	0.000000
3	#attention february #inflation #argentina lo...	0.116667	-0.083333
4	trillion m2 money stock mean worth #i...	0.393750	-0.006250
...	...	...	...
652697	kansa city area job listing non cdl truck dri...	0.000000	0.000000
652698	worker laid #employment frozen #salaries stag...	0.616667	-0.319444
652699	concept #employment involves three ingredient ...	0.000000	0.000000
652700	government cultivate lifelong #learning #minds...	0.300000	-0.050000
652701	greater st louis area job listing cdl truck ...	0.500000	0.500000

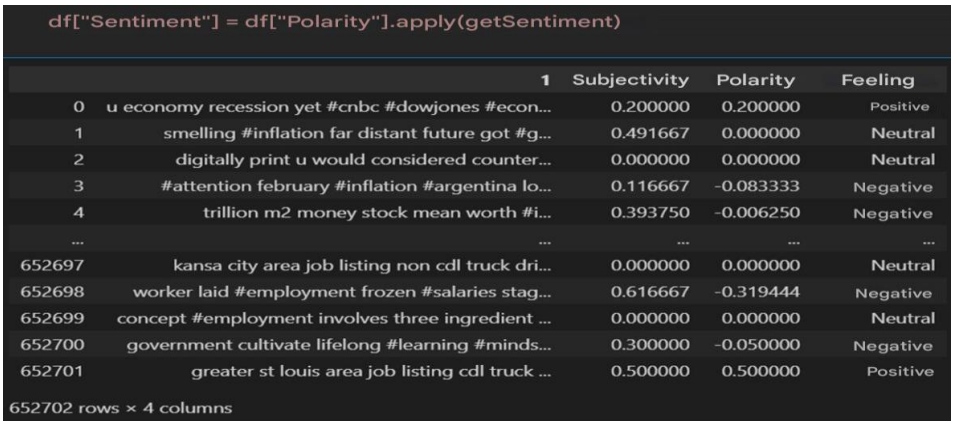
652702 rows × 3 columns

Fig. 1. Updated dataframe

With these results, the function getSentiment(score) is used, which takes as a parameter the Polarity value of the tweets to determine whether the tweet is considered positive, neutral, or negative. The function implementation is shown below:

```
def getSentiment(score):
    if score < 0: #if less than 0 it's negative
        return "Negative"
    elif score == 0:
        return "Neutral"
    else: #if greater than 0 it's positive
        return "Positive"
```

According to the above functions that produce the sentiment of tweets, this is added to the dataframe as a new column called "Feeling" (See Figure 2).



```
df["Sentiment"] = df["Polarity"].apply(getSentiment)
```

	1	Subjectivity	Polarity	Feeling
0	u economy recession yet #cnbc #dowjones #econ...	0.200000	0.200000	Positive
1	smelling #inflation far distant future got #g...	0.491667	0.000000	Neutral
2	digitally print u would considered counter...	0.000000	0.000000	Neutral
3	#attention february #inflation #argentina lo...	0.116667	-0.083333	Negative
4	trillion m2 money stock mean worth #i...	0.393750	-0.006250	Negative
...	...	...	...	...
652697	kansa city area job listing non cdl truck dri...	0.000000	0.000000	Neutral
652698	worker laid #employment frozen #salaries stag...	0.616667	-0.319444	Negative
652699	concept #employment involves three ingredient ...	0.000000	0.000000	Neutral
652700	government cultivate lifelong #learning #minds...	0.300000	-0.050000	Negative
652701	greater st louis area job listing cdl truck ...	0.500000	0.500000	Positive

652702 rows × 4 columns

Fig. 2. Creation of new "sentiment" column

The tweet dataframe must be separated into 3 different dataframes depending on their sentiment with their respective hashtag lists and a list of all downloaded tweets (tweets[]). The code for obtaining the relationship between sentiment and associated hashtag can be seen below:

```
pos_list = []
neg_list = []
neu_list = []
pos_hashtag = []
neg_hashtag = []
neu_hashtag = []
tweet_hashtags = []
positive = pd.DataFrame(columns=["Tweet", "Hashtag"])
negative = pd.DataFrame(columns=["Tweet", "Hashtag"])
neutral = pd.DataFrame(columns=["Tweet", "Hashtag"])
tweets = pd.DataFrame(columns=["Tweet", "Hashtag"])
for i in range(0, df.shape[0]): #Iterate through entire dataframe
    if df["sentiment"][i] == "Positive": #Save positive, negative and neutral tweets
        pos_list.append(df[1][i])
        pos_hashtag.append(df[2][i])
        positive = pd.DataFrame({"Tweet":pos_list, "Hashtag":pos_hashtag})
    elif df["sentiment"][i] == "Negative":
        neg_list.append(df[1][i])
        neg_hashtag.append(df[2][i])
        negative = pd.DataFrame({"Tweet":neg_list, "Hashtag":neg_hashtag})
    elif df["sentiment"][i] == "Neutral":
        neu_list.append(df[1][i])
        neu_hashtag.append(df[2][i])
        neutral = pd.DataFrame({"Tweet":neu_list, "Hashtag":neu_hashtag})
    tweet_hashtags.append(df[2][i])
positive["Hashtag"] = pos_hashtag
negative["Hashtag"] = neg_hashtag
neutral["Hashtag"] = neu_hashtag
tweets["Hashtag"] = tweet_hashtags
tweets["Tweet"] = pos_list + neg_list + neu_list
```

4.1 Dictionary Creation

Once the tweets are separated by sentiment, dictionary creation becomes possible, understanding that these dictionaries contain the most frequently used words for each sentiment. Two functions were coded to create the dictionaries:

```
sentiment_dictionary(sentiment_list,filename),
hashtag_dictionary(tweet_list,filename)
```

Parameters: a) df: name of the dataframe containing positive, neutral or negative tweets and b) filename: name of the file where the dictionary should be saved. The functions obtain the words and hashtags with their respective assignment of the most frequently used words. At this point, we can obtain the dictionary of positive tweets, negative tweets, all tweets, and hashtags.

```
sentiment_dictionary(positive,"p_all_hashtag.csv")
sentiment_dictionary(negative,"n_all_hashtag.csv")
sentiment_dictionary(tweets, "Dictall_hashtag.csv")
sentiment_dictionary(tweets, "hashtags_all_hashtag.csv")
```

Using a minimum word count threshold of 6000 for the positive tweet dictionaries (all tweets) and 2000 for the negative.csv dictionary, the dictionaries were generated. The complete dictionaries are available for reference.

5 Sentiment Analysis

In the chart shown in Figure 3, we observe the relationship between hashtags and the most recurring associated concepts. The stock market is the hashtag most frequently associated with positive concepts, as is money. Meanwhile, the inflation hashtag is primarily related to the negative aspect of high risk. On the other hand, the economy hashtag is mostly associated with necessity and government. Figure 3 also visually presents the frequency trends of words related to specific hashtags. The #inflation and #economy hashtags, being the foundation of this research, show the highest frequency of associated words. #money and #market follow in frequency. Among the associated words appearing most frequently are money, taste, necessity, exchange, and risk.

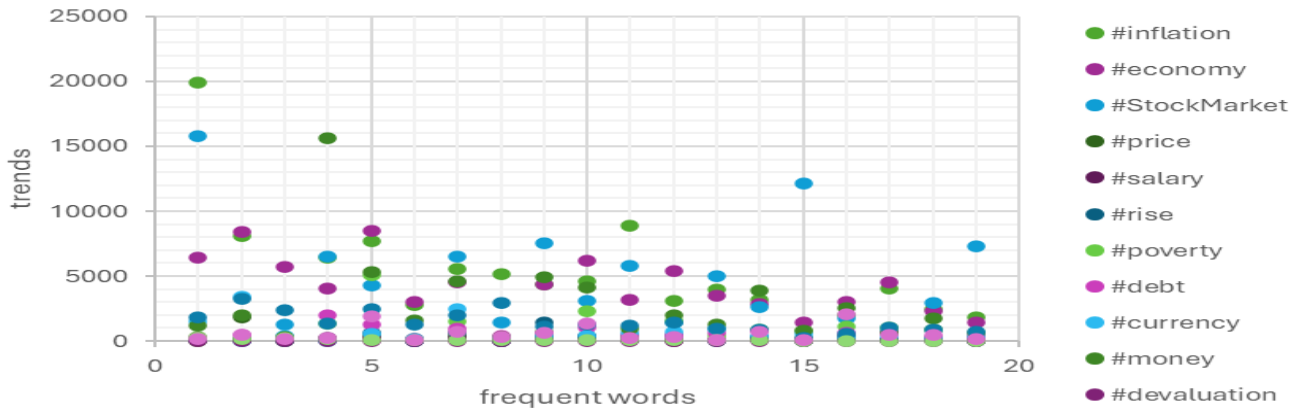


Fig. 3. Sentiment analysis chart

The charts in Figure 4 reveal the percentage of positive, negative, or neutral tweets. To analyze the polarity and subjectivity values of the tweets, it's interesting to plot these values. The red dots indicate Negative tweets, the green dots represent Positive tweets, and the blue dots denote Neutral tweets (See Figures 3 and 4). The plt.scatter() function was used to plot subjectivity on the Y-axis and polarity on the X-axis.

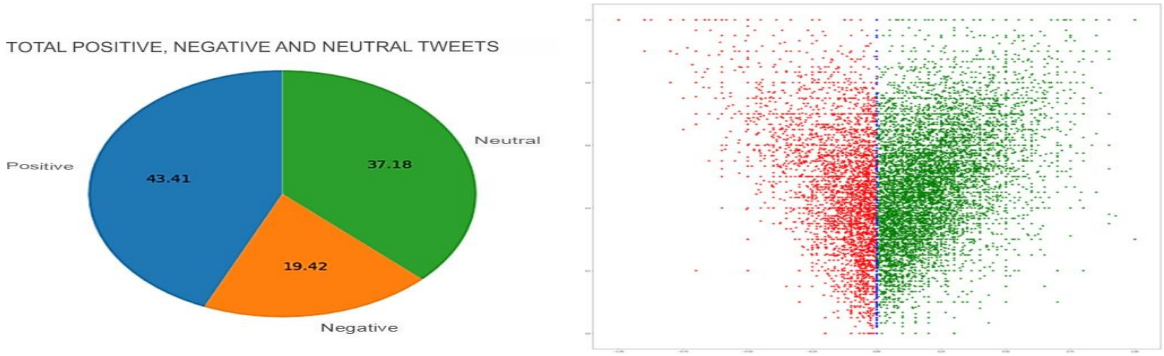


Fig. 4. Percentage charts of positive, negative, or neutral tweets and polarity charts

5.1 Analysis of Tweets by Download Date

Tweets can be processed in different ways depending on the research objectives. In this case, it's important to categorize them by download date to analyze sentiments at different stages of the pandemic. For this purpose, tweets are first sorted chronologically using the sort_values(by=0) function:

```
df = df.sort_values(by=0)
df[0] = pd.to_datetime(df[0])
```

Three masks are created to filter tweets by specific date ranges. The first mask identifies tweets posted before the pandemic, the second mask covers the first year, and the third mask represents the second pandemic year.

```
mask_before = (df[0] >= '2020-01-01') & (df[0] <= '2020-03-13')
mask_1pandemic = (df[0] >= '2020-03-14') & (df[0] <= '2021-03-13')
mask_2pandemic = (df[0] >= '2021-03-14') & (df[0] <= '2022-11-10')
before = df.loc[mask_before]
before.to_csv("before.csv")
first_year_covid = df.loc[mask_1pandemic]
first_year_covid.to_csv("first_year.csv")
second_year_covid = df.loc[mask_2pandemic]
second_year_covid.to_csv("second_year.csv")
```

It's crucial to reload the date-segmented files into separate dataframes and repeat the sentiment classification for each dataframe using the previously mentioned methods.

```
before = pd.read_csv("before.csv", header=None, usecols=[0,1])
first_year_covid = pd.read_csv("first_year.csv", header=None, usecols=[0,1])
second_year_covid = pd.read_csv("second_year.csv", header=None, usecols=[0,1])
```

After obtaining lists of positive, negative, and neutral tweets for each date-segmented dataframe, we visualized them in charts to analyze the quantity of each sentiment type during different periods. The resulting charts are present at Figure 5:

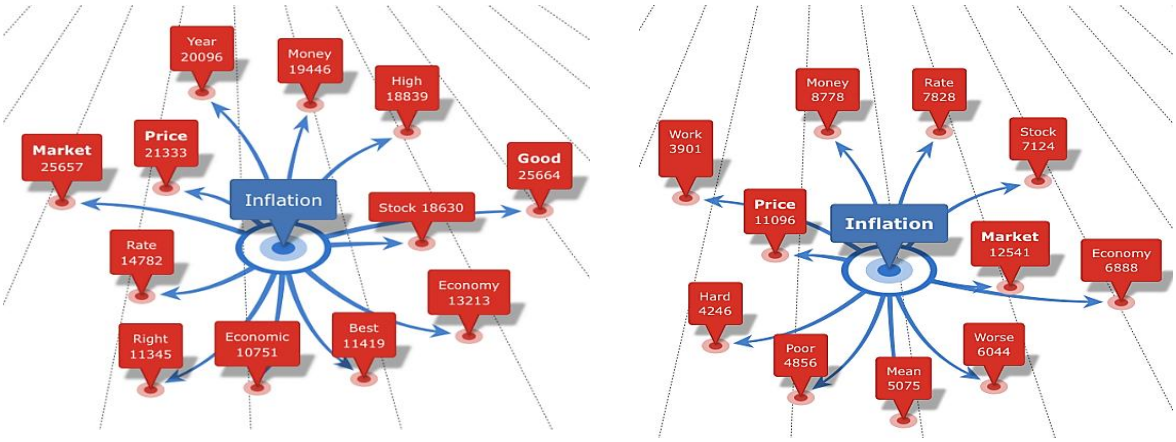


Fig. 5. Charts showing positive tweets (right) and negative tweets (left)

The charts in Figure 6 demonstrate the increase in both negative and positive tweets during the pandemic years.

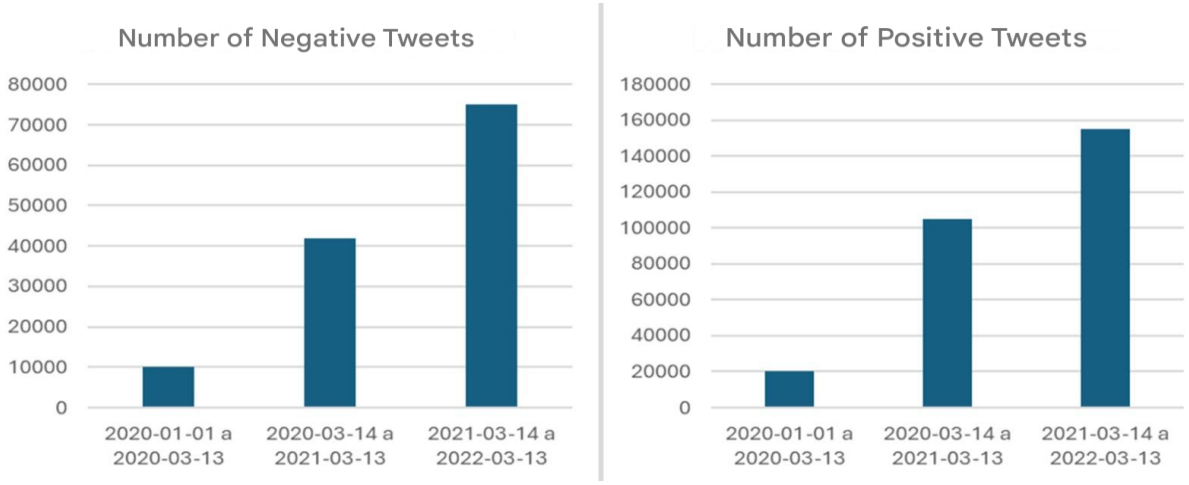


Fig. 6. Charts showing positive tweets (right) and negative tweets (left)

The final sentiment dictionary is presented in the following list (Figure 7).

	Palabras	Frecuencia	#inflation	#economy	#StockMarket	#price	#salary	#rise	#poverty	#debt	#currency	#money	#devaluation	#recession	#deflation	#employment
0	good	25664	4072	4101	7326	333	148	1450	691	574	393	4833	16	1038	67	622
1	market	25657	4064	3978	13190	191	42	83	77	235	823	1090	12	1534	60	278
2	time	22564	3745	3787	6388	289	155	731	629	605	839	3088	2	1393	75	838
3	price	21333	9911	2877	4828	1328	20	55	129	113	372	597	5	884	129	85
4	year	20096	6404	4348	3071	239	192	197	675	692	482	1773	13	1333	66	611
5	people	19736	3930	4262	2152	237	139	366	2268	654	301	3175	14	1101	46	1091
6	inflation	19478	14213	2722	633	48	20	3	67	132	201	154	10	1082	108	85
7	money	19446	3010	2024	3275	104	115	52	597	867	343	8214	14	635	57	139
8	high	18839	7281	2671	5189	267	81	114	255	405	555	808	12	935	43	223
9	stock	18630	1386	1422	14037	110	11	16	22	66	83	811	4	595	30	37
10	like	13921	2581	2269	3215	204	102	336	711	458	319	2368	4	927	37	390
11	economy	13213	2500	7877	697	24	3	11	145	214	118	138	6	1272	35	173
12	need	12678	2152	3255	1655	151	103	206	1064	476	193	2096	0	618	35	674
13	first	12000	1648	2373	3829	141	168	329	299	375	369	1435	11	652	29	342
14	best	11419	1138	1338	2379	240	59	351	246	931	615	3430	6	399	16	271
15	right	11345	2022	1894	2514	211	73	852	649	262	134	1548	3	712	32	439
16	economic	10751	2073	5641	591	53	10	12	277	187	231	146	11	1188	38	293
17	great	10739	1169	1858	2270	155	37	701	453	220	148	2562	4	703	16	443
18	higher	9350	3956	1377	2236	137	50	75	122	215	286	290	3	444	32	127
19	think	9035	1932	1800	1712	137	69	170	361	222	209	1550	2	601	38	232
20	world	8977	1495	2839	754	100	20	247	784	269	393	1198	4	657	29	188
21	better	8942	1575	2167	1589	129	90	230	518	303	182	1343	5	458	17	336
22	work	7807	851	1624	907	81	173	278	588	183	93	1543	1	274	12	1199
23	company	7268	1034	1370	2436	84	146	82	58	307	62	901	2	350	3	433
24	bank	7236	1757	1462	2058	19	34	4	174	247	438	580	3	392	20	48
25	life	7104	650	1285	813	64	67	317	682	432	76	2234	1	223	6	254
26	strong	7001	921	1316	1941	64	10	136	80	63	958	1010	2	380	15	105
27	free	6913	842	969	1238	123	35	131	563	752	279	1534	0	233	15	199

Fig. 7. Final sentiment dictionary

6 Inflation prediction

To address the prediction of monthly inflation in Mexico, two representative models were selected, with different assumptions and complexity levels. This methodological selection aimed to provide a broad spectrum of approaches, from the simplest and most transparent form to one of the most sophisticated and flexible models.

6.1 Simple Linear Regression

A deterministic model that establishes a linear relationship between time (independent variable) and monthly inflation (dependent variable). It assumes a constant rate of change. This is the most basic model, useful for identifying long-term general trends. $y_t = \beta_0 + \beta_1 t + \epsilon_t$ In this model, the observed monthly inflation in each month, denoted as y_t, is modeled as a function of time. The variable t represents the month number (for instance, January 2020 is coded as 1, February 2020 as 2, and so on). The model estimates an initial level of inflation through the intercept β_0 and a consistent monthly change through the slope β_1. Finally, the term ϵ_t accounts for the random, unexplained error each month, which is assumed to have a mean of zero.

6.2 SARIMA Model

Autoregressive integrated moving average model. This represents the most comprehensive approach in the study, incorporating seasonal components to model time series with periodic patterns. Widely used in economics, meteorology, and finance due to its ability to capture both trends and seasonal cycles.

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D X_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t \quad \textbf{(1)}$$

Where:
X_t = Observed value of the time series (monthly inflation) at timetiempo t
ϵ_t = Random error term (white noise), with zero mean and constant variance
B = Lag operator (backshift), defined as $BX_t = X_{t-1}$
s = Seasonal period (in this case, $s = 12$ for annual cycles in monthly data)

Due to the model's complexity, it must be broken down into components (First component: Non-seasonal), divided into three parts: The Autoregressive (AR) component of order p, which captures the dependence of X_t on its p past values, defined as: $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ Non-seasonal differencing, which removes non-stationary trends from the series: $(1-B)^d$ The Moving Average (MA) component of order q, which models the impact of past errors on the current value , represented as: $\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots \theta_q B^q$ The second component is the Seasonal component, also divided into three main parts. The Seasonal Autoregressive (SAR) component of order P, relating to X_t its values in the same period of previous cycles: $\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots \Phi_P B^{Ps}$ Seasonal differencing removes systematic seasonality from the series: $(1-B^s)^D$ Finally, the Seasonal Moving Average (SMA) component of order Q, which models the effect of random errors occurring in past seasonal periods on the current value: $\Theta_Q(B^s) = 1 + \Theta_1(B^s) + \Theta_2 B^{2s} + \cdots \Theta_Q B^{Qs}$

6.3 Historical Data Acquisition

The historical monthly inflation data was obtained from the National Institute of Statistics and Geography (INEGI), specifically from the National Consumer Price Index (INPC) series, publicly available on its official portal (National Institute of Statistics and Geography [INEGI], 2025). The period considered for historical analysis spans from January 2020 to May 2025 as shown in Table 1.

Table 1. Historical Inflation

X	Date	Historical Inflation	X	Date	Historical Inflation	X	Date	Historical Inflation
1	2020/01	3,24	23	2021/11	7,37	45	2023/09	4,45
2	2020/02	3,7	24	2021/12	7,36	46	2023/10	4,26
3	2020/03	3,25	25	2022/01	7,07	47	2023/11	4,32
4	2020/04	2,15	26	2022/02	7,28	48	2023/12	4,66
5	2020/05	2,84	27	2022/03	7,45	49	2024/01	4,88
6	2020/06	3,33	28	2022/04	7,68	50	2024/02	4,4
7	2020/07	3,62	29	2022/05	7,65	51	2024/03	4,42
8	2020/08	4,05	30	2022/06	7,99	52	2024/04	4,65
9	2020/09	4,01	31	2022/07	8,15	53	2024/05	4,69
10	2020/10	4,09	32	2022/08	8,7	54	2024/06	4,98
11	2020/11	3,33	33	2022/09	8,7	55	2024/07	5,57

12	2020/12	3,15	34	2022/10	8,41	56	2024/08	4,99
13	2021/01	3,54	35	2022/11	7,8	57	2024/09	4,58
14	2021/02	3,76	36	2022/12	7,82	58	2024/10	4,76
15	2021/03	4,67	37	2023/01	7,91	59	2024/11	4,55
16	2021/04	6,08	38	2023/02	7,62	60	2024/12	4,21
17	2021/05	5,89	39	2023/03	6,85	61	2025/01	3,59
18	2021/06	5,88	40	2023/04	6,25	62	2025/02	3,77
19	2021/07	5,81	41	2023/05	5,84	63	2025/03	3,8
20	2021/08	5,59	42	2023/06	5,06	64	2025/04	3,93
21	2021/09	6	43	2023/07	4,79	65	2025/05	4,42
22	2021/10	6,24	44	2023/08	4,64			

Based on this dataset, monthly inflation projections were made for the period between June 2025 and December 2028 using various enumerated time series statistical models.

6.4 Application of Linear Regression in Inflation Prediction for the Period 2025-2028

To estimate the future trajectory of monthly inflation in Mexico, a simple linear regression model was implemented, whose mathematical foundation was previously described. This model establishes a direct relationship between time (measured in months) and the observed inflation rate, under the hypothesis of a constant linear trend. In this case, the analysis was built using actual monthly inflation data for Mexico corresponding to the period January 2020 - May 2025. Each observation was coded with a temporal index X representing the sequential month number, starting with 1 for January 2020 and reaching 65 for May 2025. For the estimation, the FORECAST function in Microsoft Excel was used, which calculates the estimated value y given a new x (in this case, the future month number). This function internally implements the linear regression formula. The syntax of this function is: $FORECAST(x, known\_y; known\_x)$. In this study it was implemented as x = Future month number to project, $known\_y$ = Observed monthly inflation series (2020-2025), $known\_x$ = Temporal index corresponding to each month of the observed period

Results

The application of the FORECAST function allowed generating an extended series of monthly predictions, starting in June 2025 (X=66) and continuing through December 2028 (X=108). Table 2 of results shows the obtained outcomes.

Table 2. Inflation Forecast Using Linear Regression

X	Date	Forecasted Inflation	X	Date	Forecasted Inflation	X	Date	Forecasted Inflation
66	2025/06	5,543	81	2026/09	5,6396	96	2027/12	5,7362
67	2025/07	5,5495	82	2026/10	5,646	97	2028/01	5,7426
68	2025/08	5,5559	83	2026/11	5,6525	98	2028/02	5,749
69	2025/09	5,5623	84	2026/12	5,6589	99	2028/03	5,7555
70	2025/10	5,5688	85	2027/01	5,6653	100	2028/04	5,7619
71	2025/11	5,5752	86	2027/02	5,6718	101	2028/05	5,7683
72	2025/12	5,5817	87	2027/03	5,6782	102	2028/06	5,7748
73	2026/01	5,5881	88	2027/04	5,6847	103	2028/07	5,7812
74	2026/02	5,5945	89	2027/05	5,6911	104	2028/08	5,7876
75	2026/03	5,601	90	2027/06	5,6975	105	2028/09	5,7941
76	2026/04	5,6074	91	2027/07	5,704	106	2028/10	5,8005
77	2026/05	5,6138	92	2027/08	5,7104	107	2028/11	5,807
78	2026/06	5,6203	93	2027/09	5,7168	108	2028/12	5,8134
79	2026/07	5,6267	94	2027/10	5,7233			
80	2026/08	5,6332	95	2027/11	5,7297			

The simple linear model projects a constant and moderate growth of monthly inflation into the future, reflecting the positive slope calculated in the historical fit. This projection is deterministic and does not incorporate seasonal effects or external shocks,

so it is interpreted as a baseline scenario of prolonged linear trend. The advantage of this approach lies in its simplicity and transparency, being easy to apply and interpret. However, its limitations are clear:

- Does not capture nonlinear patterns (curves or variable accelerations over time)
- Ignore seasonal behaviors that may repeat cyclically
- Assuming the relationship between inflation and time remains constant in the future

For these reasons, in the comparative analysis of the study, these results are contrasted with more sophisticated models, such as Polynomial Regression and SARIMA, which can capture dynamic and seasonal components present in the inflation series.

6.5 Application of the SARIMA Model in Inflation Prediction for the Period 2025-2028

To estimate monthly inflation in Mexico and project it through December 2028, a SARIMA (Seasonal AutoRegressive Integrated Moving Average) model was employed. This model is widely used in time series analysis with seasonal patterns and non-stationarity. The SARIMA model extends the well-known ARIMA by incorporating seasonal terms, which is particularly useful when data shows periodic fluctuations (e.g., annual cycles). Its general form is expressed as:

$$SARIMA(p,d,q) \times (P,D,Q,s) \tag{2}$$

In this study it was implemented as p = non-seasonal autoregressive (AR) order, d = Degree of non-seasonal differencing q = Non-seasonal moving average (MA) order, P = Seasonal autoregressive order, D = Degree of seasonal differencing, Q = Seasonal moving average order, s = Seasonal periodicity (in this case, 12 months)
This model combines seasonal and non-seasonal components to fit the complex dynamics of the series, including effects such as trend, annual cycles, and transient shocks. For the implementation of this model, a Python script was developed using the historical data previously obtained from INEGI (National Institute of Statistics and Geography [INEGI], 2025), covering January 2020 to May 2025 (Table 1).

Before fitting an SARIMA model, it is essential to verify the stationarity of the time series, meaning that its statistical properties (mean, variance) remain constant over time. For this purpose, the Augmented Dickey-Fuller (ADF) test was used, which tests the null hypothesis that the series has a unit root (non-stationary), presented as:

$$H_0: The\ series\ is\ non-stationary \tag{3}$$

$$H_1: The\ series\ is\ stationary$$

The following code block allowed us to perform this test using the statsmodels.tsa.stattools library:

```
result = adfuller(timeseries.dropna())
```

The obtained results gave us an ADF Statistic: -1.55 and a p-value: 0.51. Since the p-value > 0.05, we fail to reject the null hypothesis of a unit root in , indicating the need to difference the series. This differencing step was handled automatically during the optimal parameter search. To objectively evaluate the model's predictive capability, the historical data was split into two sets: a training set comprising 80% of the data (January 2020 - April 2024) and the remaining 20% was used for model validation. The selection of parameters and was performed automatically using the auto_arima algorithm, which optimizes the orders and differences based on statistical criteria, facilitating the optimal fitting of the SARIMA model to the historical data. This optimization process is implemented in the following line of code:

```
auto_model = auto_arima(
    train,
    seasonal=True,
    m=12,
    trace=True,
    error_action='ignore',
    suppress_warnings=True,
    stepwise=True
)
```

The optimal model that best fit the historical data was: SARIMA(1,1,0)×(2,0,0,12). The SARIMAX model was fitted and used to generate predictions on the validation set, as shown below:

```
# Model fitting
model = SARIMAX(train, order=order, seasonal_order=seasonal_order)
model_fit = model.fit(disp=False)
# Prediction on test set
forecast_test = model_fit.get_forecast(steps=len(test))
forecast_values = forecast_test.predicted_mean
conf_int = forecast_test.conf_int()
```

Once the prediction on the test set was generated, we proceeded with forecasting through December 2028:

```
# Future prediction until December 2028
future_steps = (2028 - 2025) * 12 + (12 - df.index[-1].month)
forecast = model_fit.get_forecast(steps=future_steps)
future_values = forecast.predicted_mean
future_conf_int = forecast.conf_int()
```

An essential characteristic of the SARIMA model is its ability to generate confidence intervals around point predictions, expressed as

$$\hat{y}_{t+h} \pm z_{1-\frac{\alpha}{2}} * \hat{\sigma}_h \tag{4}$$

In this study it was implemented as follows $\hat{y}_{t+h}$ = Point prediction for steps ahead, $z_{1-\frac{\alpha}{2}}$ = Critical value from normal distribution, $\hat{\sigma}_h$ = Accumulated prediction standard error at horizon h

These intervals allow expressing uncertainty in projections, particularly relevant for economic series with significant variability. The predictions made by the SARIMAX model include 95% confidence intervals, which were generated through the following line: future_conf_int = forecast.conf_int().
To quantitatively evaluate the predictive capability of the SARIMA model, its predictions were compared with actual values from the validation set (equivalent to the most recent 20% of historical data). Three standard time series analysis metrics were used:

- MAE (Mean Absolute Error): Measures the average absolute error between actual values and predictions
- RMSE (Root Mean Squared Error): More severely penalizes large errors by squaring differences before averaging and then taking the square root
- MAPE (Mean Absolute Percentage Error): Calculates the mean absolute percentage error, useful for comparing different scales by expressing relative error as percentage

These metrics were calculated in Python with the following lines:

```
# Accuracy metrics
mae = mean_absolute_error(test, forecast_values)
rmse = np.sqrt(mean_squared_error(test, forecast_values))
mape = np.mean(np.abs((test - forecast_values) / test)) * 100
```

The results obtained for the validation set were as follows:

Table 3. Model accuracy metrics

Metric	Value	Interpretation
MAE	1,223967	Average absolute error between predictions and actual values
RMSE	1,455869	Root mean squared error (gives more weight to large errors)
MAPE	29,93025	Average percentage error (useful for comparing different scales)

The MAE, with a value of 1.22 percentage points, indicates that on average the model's predictions deviate from observed values by approximately one percentage unit, representing reasonable precision given the typical volatility of monthly inflation. The RMSE, with a slightly higher value of 1.46, more severely penalizes larger errors, showing that occasionally the model may generate more significant deviations from observed values. Finally, the MAPE, which quantifies relative error expressed as a percentage, reached 29.93%, indicating the model maintains an error margin close to 30% relative to actual values. This percentage, while potentially considered high in other contexts, is coherent and expected when predicting complex economic

variables like inflation, where external factors and unmodeled shocks affect estimation accuracy. Collectively, these metrics demonstrate balanced performance, highlighting the model's ability to capture general trends and seasonality, albeit with inherent limitations due to the stochastic nature of the phenomenon.

The projections generated by the model include point estimates of monthly inflation and their respective 95% confidence intervals, shown in the following results table:

Table 4. Inflation Forecast Using SARIMA Model

Date	Inflation Prediction	Lower limit	Upper limit	Date	Inflation Prediction	Lower limit	Upper limit
2025/06	4,903979	4,21376	5,594198	2027/04	5,89882	1,684843	10,1128
2025/07	5,293889	4,090754	6,497025	2027/05	5,759343	1,528902	9,989784
2025/08	5,411069	3,772923	7,049216	2027/06	5,672803	1,420018	9,925589
2025/09	5,298797	3,288189	7,309405	2027/07	5,499716	1,221897	9,777535
2025/10	5,428696	3,092893	7,764499	2027/08	5,43819	1,13424	9,74214
2025/11	5,671532	3,046078	8,296985	2027/09	5,387942	1,057481	9,718403
2025/12	5,865737	2,977798	8,753675	2027/10	5,348943	0,991902	9,705984
2026/01	5,622995	2,493821	8,752169	2027/11	5,320174	0,936616	9,703732
2026/02	5,436149	2,082772	8,789525	2027/12	5,353797	0,943839	9,763755
2026/03	5,879906	2,316293	9,443518	2028/01	5,422653	0,986435	9,85887
2026/04	6,163866	2,401702	9,92603	2028/02	5,464509	1,002178	9,92684
2026/05	6,237	2,286232	10,18777	2028/03	5,376456	0,88816	9,864751
2026/06	6,22009	2,202578	10,2376	2028/04	5,407521	0,893409	9,921633
2026/07	6,252361	2,200652	10,30407	2028/05	5,475599	0,935817	10,01538
2026/08	6,275822	2,2005	10,35114	2028/06	5,541928	0,959554	10,1243
2026/09	6,411398	2,316422	10,50638	2028/07	5,649031	1,015928	10,28213
2026/10	6,395133	2,282119	10,50815	2028/08	5,682465	0,995341	10,36959
2026/11	6,300842	2,1705	10,43118	2028/09	5,664748	0,922552	10,40694
2026/12	6,143465	1,996132	10,2908	2028/10	5,697982	0,900633	10,49533
2027/01	6,179681	2,015539	10,34382	2028/11	5,754324	0,902146	10,6065
2027/02	6,223688	2,042852	10,40452	2028/12	5,791957	0,885435	10,69848
2027/03	6,102737	1,905294	10,30018				

These predictions reflect the expected evolution of monthly inflation in Mexico for the period between June 2025 and December 2028. The point estimates of predicted inflation range approximately between 5.3% and 6.4% on average annually, showing a relatively stable trend with slight fluctuations throughout the analyzed horizon. The associated confidence intervals progressively widen over time, a characteristic phenomenon in time series reflecting accumulating uncertainty as projections move further from the observed period. This characteristic is particularly relevant in economic analysis, where variables like inflation are exposed to external shocks, macroeconomic cycles, and unexpected events that make absolute precision in long-term projections difficult. The above table provides not only point estimates but also upper and lower limits with statistical confidence levels, offering a comprehensive and prudent view of forecasts.

To illustrate the model's performance, we generated a chart for visual interpretation of the results, composed of the following elements:

- Training data (blue): historical series used to fit the model
- Validation data (green): actual values reserved for testing
- Validation prediction (red dotted): model evaluation on unseen data
- Future prediction (purple dotted): projection through December 2028
- Confidence intervals (pink area): quantification of associated uncertainty

The chart was generated in Python during code execution, with the resulting chart shown below (see Figure 8):

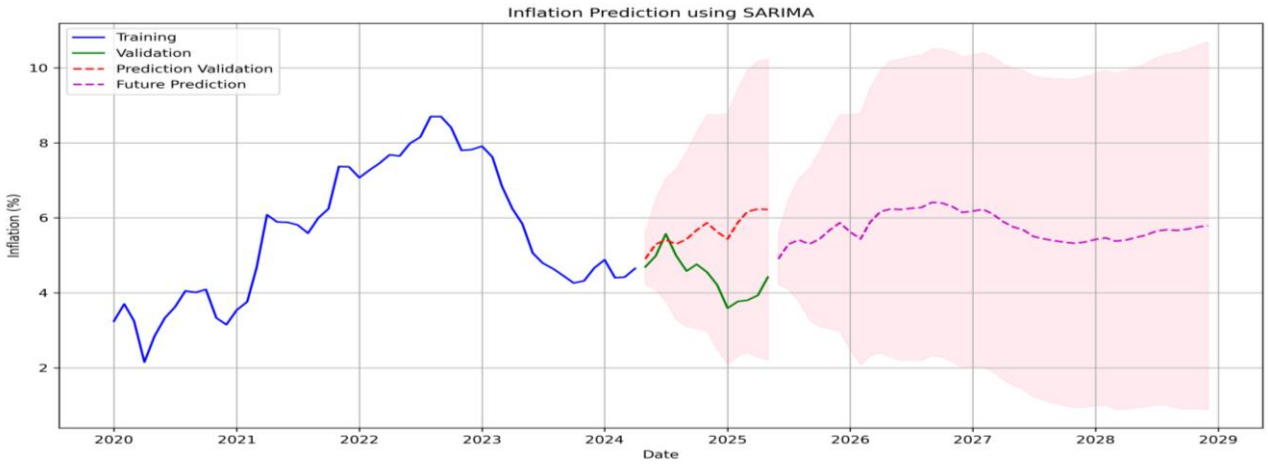


Fig. 8. SARIMA Model Prediction Chart

The chart shows that the model successfully captures the general dynamics and seasonality of inflation during the validation period, with reasonable errors relative to actual data. The future projection suggests a slightly fluctuating but relatively stable trajectory between 2025 and 2028, with confidence intervals that widen as the horizon progresses, reflecting the increasing uncertainty inherent in long-term economic predictions. To facilitate a medium-term interpretation from a macroeconomic perspective, the monthly predictions were aggregated by year. The following table presents the estimated annual average inflation along with the average values of the lower and upper confidence interval limits:

Table 5. Annual Summary of Estimated Inflation with SARIMA

	Estimated Inflation			Lower Limit	Upper Limit
Year	Mean	Min	Max	Mean	Mean
2025	5,410529	4,903979	5,865737	3,497485	7,323572
2026	6,111586	5,436149	6,411398	2,24581	9,977361
2027	5,682153	5,320174	6,223688	1,406952	9,957354
2028	5,577431	5,376456	5,791957	0,940632	10,21423

The annual analysis reveals relatively stable average inflation throughout the projected period, with values between 5.4% and 6.1%. However, the confidence interval limits reflect the breadth of possible scenarios: the minimum values decrease slightly over time, suggesting some margin for inflation reduction; in contrast, the upper limits remain high, revealing a persistent probability of high inflation scenarios. This comprehensive view underscores the importance of incorporating confidence bands in economic decision-making, as they allow anticipating realistic variation ranges rather than just point estimates. In Mexico, internal and external factors can rapidly alter the course of macroeconomic variables - this probabilistic approach provides a more solid foundation for planning, risk assessment, and policy formulation.

7 Discussion

This study complements and contrasts with the existing literature in several key aspects. Whereas Muller Durán (2022) and the Bank of Mexico (2020) identified supply/demand imbalances and inflationary trajectories using macroeconomic models, our sentiment analysis of Twitter captured the underlying public perception of these phenomena in real time, providing qualitative validation for their quantitative findings. Our research corroborates the thesis of Nava Olivares (2020) regarding the heterogeneity of perception across socioeconomic strata; however, our sentiment lexicons demonstrate that inflation concern was a transversal sentiment, linguistically manifested through lexical units such as "peor" (worse) and "alto" (high) within semantically negative contexts. From a methodological perspective, while Li and Tang (2022) also employed Twitter data and architectures such as BERT, our implementation utilizing TextBlob and a Naive Bayes classifier demonstrated comparable effectiveness while offering greater computational accessibility. The principal innovation of this work lies in the hybrid integration of sentiment analysis with forecasting models (SARIMA), thereby combining the immediacy of Natural Language Processing (NLP) with the statistical rigor of time series analysis.

In contrast to the 8% inflation projection by García Pérez and Mendoza Rivera (2025) using a Bayesian expert aggregation model, our SARIMA-based projections yield more moderate estimates (5.3%-6.4%). This discrepancy underscores how divergent methodologies and data sources (expert surveys versus historical data) can generate distinct scenarios, thereby enriching the analytical landscape. This work demonstrated the efficacy of NLP-based sentiment analysis for monitoring public perception regarding inflation, validating that contextual semantics are crucial for accurate classification (e.g., the negative connotation of "high" in economic contexts). The primary contribution of this research is the development of a dual methodological framework that integrates qualitative sentiment analysis with quantitative time series models, thereby offering a versatile tool for socioeconomic analysis.

8 Conclusions

Currently, sentiment analysis through NLP is highly useful for studying public perception on social networks like Twitter and provides a timely method to analyze opinion topics such as the focus of this work - inflation perception during the COVID period. Throughout the study, we observed that the most frequently used words in negative tweets were market, price, inflation, high, rise, food, poor, dollar, worse. These extracted words are linked to problems of rising unemployment and prices, coupled with food demand. We also found words like worse and hard, which are clearly considered negative, indicating that the sentiment classifier functions correctly. Similarly, we can conclude that the classification correctly resolved the sentiment analysis when observing positive tweet charts with words like higher, better, strong, love, and good - words considered positive according to language dictionaries. However, although positive words like high and higher are normally considered positive, in our working context they are not, as they may indicate high prices, high inflation, etc. This makes sense, as when observing positive and negative tweet charts across different dates, we see that while negative tweets increase, so do positive ones - there are even more positive than negative tweets. Additionally, when reviewing positive tweet dictionaries, we can see that words like high and higher have higher frequency in words like inflation, economy, and Stock Market. Therefore, we can conclude that although some frequently repeated words may inherently have positive sentiment, when used in another context they can carry negative sentiment. In this scenario, our methodology is consistent, adequately resolves sentiment analysis, and the developed code is openly available via links for use in other problems.

On the other hand, the analysis of monthly inflation in Mexico between January 2020 and May 2025, based on INEGI data, enabled forecasting for the period between June 2025 and December 2028 using two mathematical tools: linear regression and the SARIMA model. With linear regression, we observed a reasonably constant growth trend with approximate monthly inflation of 5.5%. However, this projection is limited as it doesn't consider seasonal and external factors. Regarding the SARIMA model that includes these dynamic factors, it presents a more precise perspective by capturing inflation seasonality and volatility, allowing the establishment of confidence intervals with more reliable and broader estimates, while also reflecting uncertainty in future projections. The SARIMA model's accuracy metrics demonstrate good performance in inflation prediction with a 29.93% margin of error. The predictions achieved indicate prudently stable inflation, with projections varying between 5.3% and 6.4% annually. The confidence interval limits underscore the importance of acknowledging uncertainty in economic decisions, as unforeseen factors could alter inflation's course. In summary, the analysis highlights the need to adopt a probabilistic approach in economic policy formulation, offering a more realistic basis for planning and risk assessment in a dynamic economic environment.

References

Banco de México. (2020). *Evolución de la inflación en distintos países en el contexto de la pandemia de COVID-19*. Contraste Regional CIISDER, 8(16).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media. https://www.nltk.org/

Centers for Disease Control and Prevention. (2022, August 16). *CDC Museum COVID-19 timeline*. https://www.cdc.gov/museum/timeline/covid19.html

Detmers, G.-A., Ho, S.-J., & Karagedikli, Ö. (2022). Understanding consumer inflation expectations during the COVID-19 pandemic. *Journal of Monetary Economics, 130*, 1–16. https://doi.org/10.1016/j.jmoneco.2022.06.002

García Pérez, L. E., & Mendoza Rivera, R. J. (2025). Modelación de expectativas de inflación en México: Una perspectiva mediante inferencia bayesiana. *Análisis Económico, 40*(104), 29–47. https://analisiseconomico.azc.uam.mx/index.php/rae/article/view/1361

GeeksforGeeks. (2025, July 23). *Twitter sentiment analysis on Russia-Ukraine war using Python*. https://www.geeksforgeeks.org/twitter-sentiment-analysis-on-russia-ukraine-war-using-python/

GeeksforGeeks. (2025, July 9). *Twitter sentiment analysis using Python*. https://www.geeksforgeeks.org/python/twitter-sentiment-analysis-using-python/

Instituto Nacional de Estadística y Geografía. (2025). *National Consumer Price Index (NCPI): Base second half of July 2018; update of basket and weights 2024*. https://www.inegi.org.mx/temas/inpc/

JustAnotherArchivist. (n.d.). *snscrape* [Software]. https://github.com/JustAnotherArchivist/snscrape

JustAnotherArchivist. (n.d.). *snscrape*: A Python library for scraping social networking sites. https://github.com/JustAnotherArchivist/snscrape

Loria, S. (n.d.). *TextBlob: Simplified text processing* [Software]. https://textblob.readthedocs.io/en/dev/

Morales Pelagio, R. C., Robles Ulloa, O., & Mora Gutiérrez, A. A. (2024). Comparación de la inflación-desempleo de Estados Unidos y México al inicio de la pandemia de COVID-19. *ACADEMO, 11*(3), 261–270. https://doi.org/10.30545/academo.2024.set-dic.5

Muller Durán, N. I. (2022). *Impactos económicos del COVID-19 en la inflación* (Proyecto PAPIIT IA301621). Universidad Nacional Autónoma de México. http://www.economia.unam.mx/assets/pdfs/econmex/07/03%20Nancy%20Muller.pdf

Nava Olivares, R. (2020). Percepción social de la era post COVID-19. *Contraste Regional, 8*(16), 57–78.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* (pp. 79–86). Association for Computational Linguistics. https://doi.org/10.3115/1118693.1118704