# Sentiment Analysis Report

Group members:

Andrés Vizcaya Santacruz

Lesly Andrea Suárez Sánchez

Course: Artificial Inteligence

Dr. Alejandra Hernández Sánchez

Due date: May 15th 2025

## Contents

# 1 Introduction

Tourism is a growing contributor to Mexico's PIB, representing 8.6% of the total PIB in 2023, a 4.4% growth from 2022 [1]. Specifically, "Pueblos Mágicos" have had an impact on the economy of small villages in México. Tourism represents 13.5% of the economy of small villages in Mexico [2]. In this context, it's imperative to find ways to improve this sector. Online reviews of "pueblos mágicos" can be significant in analyzing the satisfaction of clients during their stay in Mexico. However, classifying these Reviews by hand can time consuming and unreliable.

Therefore, this project will apply a hybrid model composed of a neural network with a non-linear classification model to predict the satisfaction of tourists on a sacle of 1-5 and classify the review into on of three categories; Hotel, Attractive and Restaurant.

# 2 Literature Review

### Recurrent Neural Networks for Sentiment Analysis

RNNs are a class of neural networks designed to handle sequential data by maintaining a hidden state that captures information from previous time steps. This property makes them suitable for text classification tasks, including sentiment analysis. Variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been widely adopted to overcome the vanishing gradient problem inherent in traditional RNNs [3].

Several studies have demonstrated the effectiveness of RNN-based models in sentiment analysis tasks, particularly when using pre-trained word embeddings such as GloVe or Word2Vec to initialize the input representations [4]. These models are capable of capturing complex linguistic patterns and dependencies, leading to more accurate sentiment classification.

In the box below, the mathematical formulation of the RNN update rule is presented, illustrating how the hidden state is iteratively updated at each time step to incorporate new input information while preserving historical context.

**Recurrent Neural Network Update Rule**

Given a sequence of words represented as vectors $x_1, x_2, \ldots, x_T$, the RNN updates its hidden state $h_t$ at each time step $t$ as follows:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

Where:

- $W_{hx}$ and $W_{hh}$ are the input and recurrent weight matrices.
- $b_h$ is the bias vector.
- $f$ is typically a non-linear activation function such as tanh or ReLU.

### Combining RNN with SVM or Regression Models

While RNNs excel at feature extraction from sequential data, some studies have explored hybrid approaches where the features extracted by RNNs are fed into traditional machine learning models such as SVM or regression algorithms. This approach aims to leverage the representational power of deep learning while benefiting from the robustness and generalization capabilities of classical algorithms.

After extracting the final hidden state $h_T$ from the RNN, this vector can be treated as a feature representation $\phi(x)$, which is fed into an SVM classifier:

**SVM Classification**

The extracted feature vector $\phi(x)$ from the RNN is classified using an SVM decision function:

$$y = \text{sign}(\mathbf{w}^T\phi(x) + b) \tag{2}$$

Where:

- $\mathbf{w}$ is the weight vector learned by the SVM.
- $b$ is the bias term.

For instance, Wang et al. (2016) proposed a method where deep features obtained from an LSTM model were used as input to an SVM classifier, achieving better performance compared to using either method independently [5]. Similarly, using regression models on RNN-generated embeddings allows for fine-grained sentiment scoring, such as predicting ratings on a numerical scale, which can be more informative than simple categorical sentiment classification.

The combination of RNNs and SVMs or regression models has shown to be a promising direction, particularly in scenarios where the interpretability and generalization of classical models are desired alongside the deep feature extraction capabilities of neural networks. However, challenges such as data preprocessing, imbalance handling, and efficient feature extraction remain critical factors affecting the success of these hybrid systems.

In summary, the integration of RNN-based architectures with SVMs or regression models offers an effective approach for sentiment analysis tasks, particularly in domains with complex and nuanced opinions. Future research may focus on optimizing these pipelines and exploring transfer learning strategies to enhance performance further.

# 3 Methodology

## 3.1 Preprocessing

The dataset consisted of TripAdvisor 208,051 reviews put into six columns.

- Title of Review
- Review
- ”Pueblo Mágico”
- Type of attraction reviewed
- Sentiment Polarity

In order to make the classification by Type of attraction smoother a mew column was created. This column mapped the type of attraction to a number.

- Restaurant -> 1
- Hotel -> 2
- Attractive -> 3

With a quick analyisis to the dataset two reviews without a title were identified. This was fixed in the Model Develpoment.

## 3.2 Model Development

### 3.2.1 Pre-processing texts

As we are working with written reviews, it is important to normalize the text. In this case, normalization means turning all of our text into lowercase letters. Additionally, the column of Title and

Review were merged into a single column. After making our data homogenous, filler words and punctuation signs were eliminated out with the help of the ”spacy” tokenizer.

As the reviews varied in the use of accents and special characters in representation of the accents it was decided to eliminate the special characters to achieve a more managable dataset.

The data set was then split into two new datasets with scikit learn. 80% of the dataset was saved for traning and th remaining 20% for testing.

To avoid the data leakage problem the training data set was inspected for class imbalance and corrected only in the training dataset. The texts classified with a polarity of one had 4353 observations, meanwhile, the texts with a polarity of five had 109,248. In order to increase the chances of a working model for all classes we reduced the size of every class to 4,300 observations. This was done as to not contaminate the original dataset with information generated by other artificial intelligent models and be a data set comprised soley from human reviews. In other words, undersampling was applied to balance the dataset.

Finally the texts were vectorized with TfidfVectorizer.

### 3.2.2 Model for Polarity

In order to create a model to predict the polarity of a text we used the training and testing datasets. Starting with a classic Bayessian classifier. The Naive Bayes classifier had a poor preformance with

| Model | Accuracy | Weighted avg Precision |
|---|---|---|
| Naive Bayessian | 0.66 | 0.51 |
| Logistic Regression | 0.64 | 0.70 |
| Decision Trees | 0.44 | 0.59 |
| Random Forest | 0.61 | 0.67 |
| SVM | 0.59 | 0.74 |

Table 1: Table comparing classification models by the variable ’Polarity’

Table 1 shows us the comparison between different classifiers. It’s important note that the parameters for the Decision Trees and SVM to obtain these results were gathered through optimizing the parameters with Random CV search with 5-Fold cross validation. The obtained hyperparameters were then manually imputeed into our models.

| Modelo | Hiperparámetros |
|---|---|
| Logistic Regression | C:1 |
| Decision Tree | max_depth:30 ; min_samples_split: 28 |
| Random Forest | max_depth:40 ; min_samples_split: 32 ; n_estimators: 100 |

Table 2: Optimized Hyperparameters for the ”Polartiy” model

Once the final model was selected, we proceeded to develop a hybrid approach consisting of an RNN combined with an SVM classifier using optimized parameters to build the sentiment analysis model. The neural network had the following characteristics:

- LSTM: 128
- dropout: 0.3
- LSTM(recurrente): 64
- dropout: 0.3
- dense layer (softmax): 5

This RNN model is builty with a loss function of sparse_categorical_crossentropy and the optimization function corresponds to ’Adam’.

#### 3.2.3 Model for Type of attraction

The dataset used to classify the texts bassed on the type of attraction was different from that of the Model for Polarity. This is because the classes bassed on the type of attraction were balanced in our training dataset therefore, there was no need to balance the dataset.

| Modelo | Hiperparámetros |
|---|---|
| Logistic Regression | C:1 |
| Decision Tree | max_depth:10 ; min_samples_split: 42 |

Table 3: Optimized Hyperparameters for the ”Type” model

Consequently, and due to the stark differences in reviews for each type, the results obtained from Bayesian models and linear models were quite remarkable. This can be observed in...

| Model | Accuracy | F1-score |
|---|---|---|
| Naive Bayessian | 0.94 | 0.94 |
| Logistic Regression | 0.96 | 0.96 |
| Decision Trees | 0.90 | 0.90 |

Table 4: Table comparing classification models by the variable 'Type'

# 4 Results

Linear and probabilistic classifiers showed good performance, with Logistic Regression standing out as the best option due to its balance between accuracy and interpretability.

On the other hand, given the increased complexity (five classes instead of binary or simple multiclass categories), linear classifiers did not achieve optimal performance. It was necessary to implement nonlinear models (such as Random Forest and neural networks), which are better suited to capturing the complex relationships present in textual data.

## 4.1 Model Evaluation

The final hybrid model had a poor preformance in the category for 5 star reviews. Mostly confusing them for 4 star reviews. Throug Figure 2 we can observe the difference in learning for each category. It seem the class for 1 star reviews is th category with the most learning out of all of them. Meanwhile categories 4 and 5 have the worst preformance from all.

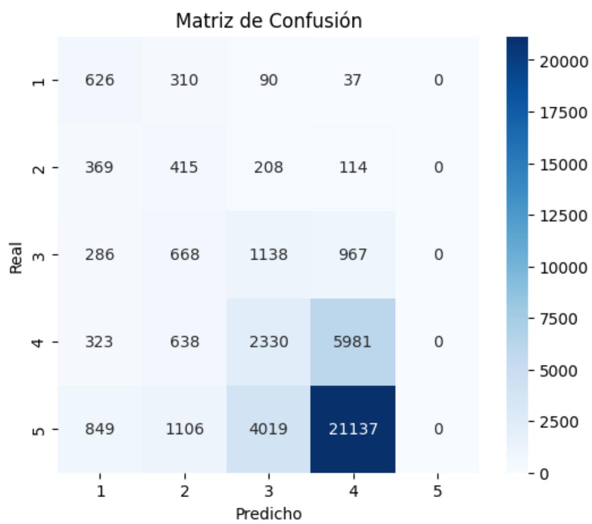


Figure 1: Confusion Matrix hybrid model

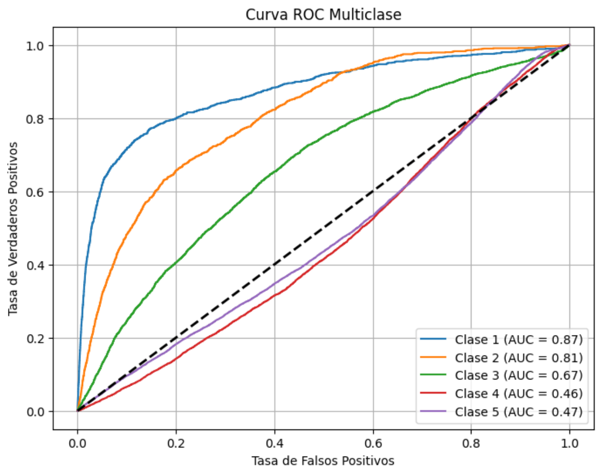


Figure 2: ROC curve hybrid model

On the other hand, dude to the nature of differences in the metadata for the type of attraction being reviewd, we see much better esults for the type model. Specifically our class 1 has the most amount of true positives. All threes categories seem to do well in the learning phase of the modelo; Figure 4

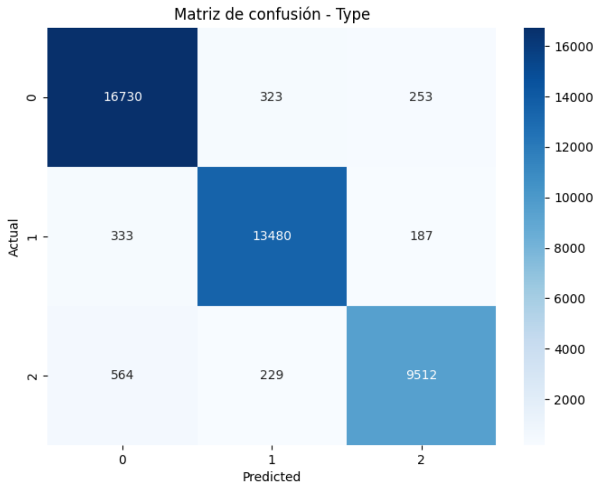


Figure 3: Confusion Matrix model ”Type”

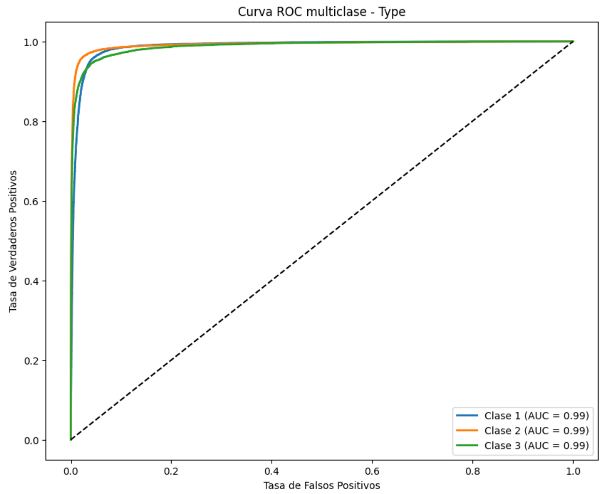


Figure 4: ROC curve model ”Type”

## 5 Aplication

This model was saved and used in a GUI, built with `tkinter` from `Python` with a textbox to input a review from the user and returns a predicted polarity of ”Very negative”,”negative”,”neutral”,positive” and ”very positive”. It also returns the type of attraction the review is refering to including ”hotel”, ”restaurant” and ”attracive”. Figure 5 shows an example of the application.

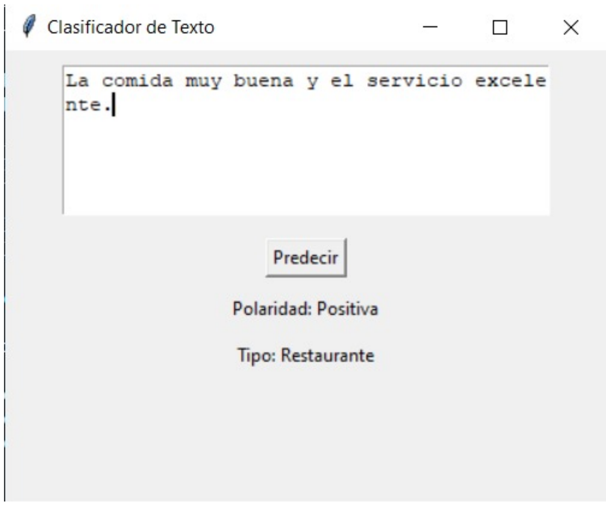


Figure 5: Application of the hybrid and ”type” model with user review

## 6 Conclusions

In basic categorization tasks, such as classifying types of establishments (restaurant, hotel, store, etc.), linear models like Logistic Regression are usually sufficient and highly efficient. This is because, in these scenarios, the relationships between the input features and the categories are relatively straightforward and can be well captured using linear functions. Additionally, the simplicity of these models allows for clear interpretation of the results, as well as quick implementation and tuning.

However, when the problem becomes more complex, such as in the case of fine-grained sentiment analysis—where the goal is not only to distinguish between positive or negative sentiments but also among multiple classes or levels (e.g., a scale from 1 to 5)—the relationships between variables become much more intricate and nonlinear. In these cases, linear models tend to fall short, as they lack the capacity to capture complex patterns or nonlinear interactions between features. Therefore, it becomes necessary to use more sophisticated and flexible algorithms, such as tree-based models, SVMs with nonlinear kernels, or neural networks, which can model more complex relationships and handle the variability and nuances present in the data more effectively.

## References


[1] Instituto Nacional de Estadística y Geografía (INEGI). (2024). Porcentaje y variación anual. https://www.inegi.org.mx/temas/turismosat/

[2] Secretaría de Turismo (SECTUR). (2024, abril 9). El turismo representa el 13% de la economía de los municipios con Pueblos Mágicos. https://www.gob.mx/sectur/prensa/el-turismo-representa-el-13-de-la-economia-de-los-municipios-con-pueblos-magicos

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26.

[5] Wang, X., Li, W., Wang, M., & Zhang, W. (2016). Combining deep learning and SVM for sentiment analysis. Neurocomputing, 210, 227-233.

[6] Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O’reilly.

[7] Jurafsky, D., & Martin, J.H. (2024). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. Stanford.