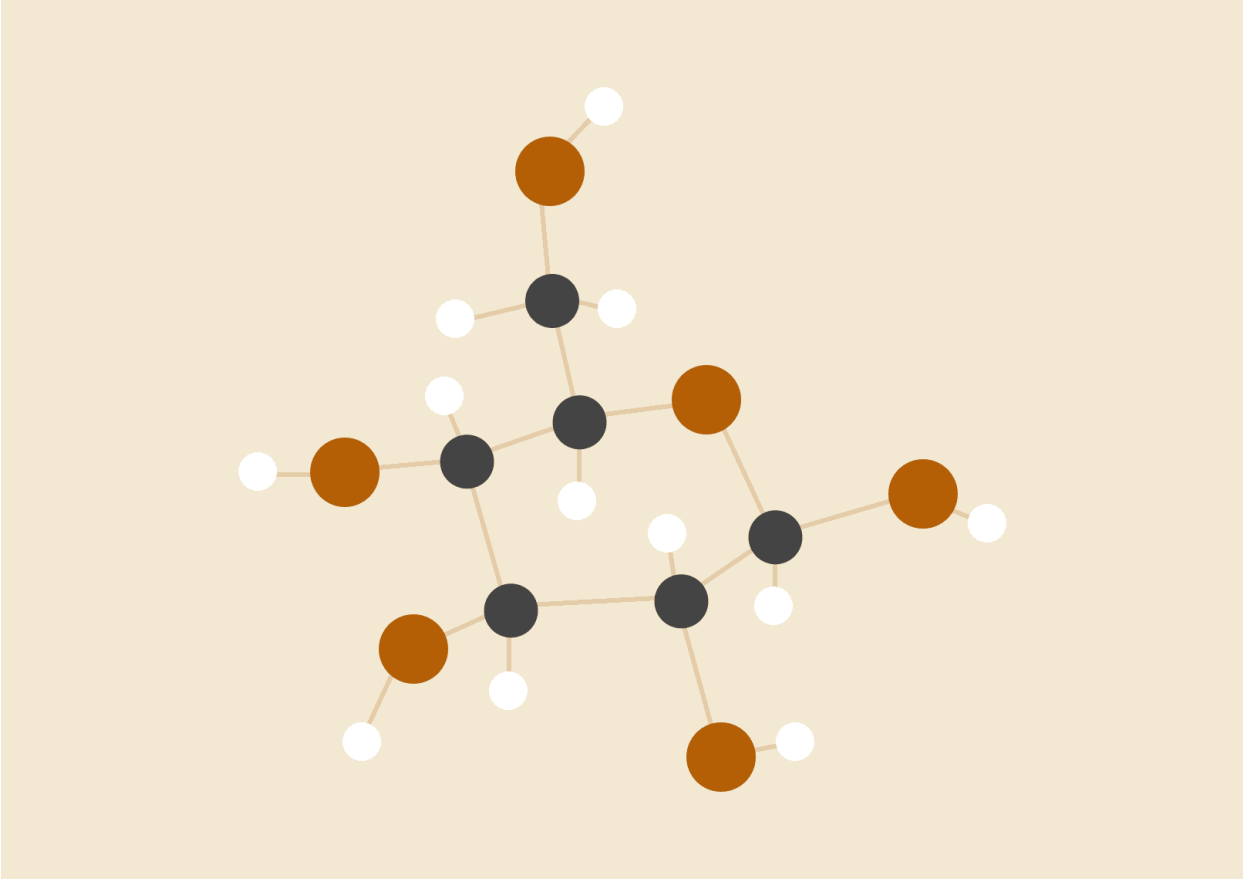
Insights and observations

CLA Tech Interview



CHERIFI Imane

21/09/2022

IMDB Reviews Dataset

From the EDA performed on the dataset we noticed the following:

- **The dataset is balanced**: the number of positive reviews is equal to the number of negative reviews (**12499**) in both datasets (**training** set and **test** set).
- There is no correlation between the size of the comments and the sentiment of the consumer, since we can find short and long reviews when it is the case of a positive or negative sentiment. However, the longest review is positive.
- The top 20 most repeated words in positive and negative comments are very similar. The only difference is in the words : “great”, “well”, “also”and “life” which are used in positive comments and the words: “bad”, “don’t”, “much”, “people” and “thing” which are commonly used in negative comments.

Problem (Task)

The task is to perform a sentiment analysis on the IMDB Review Dataset. It is a Binary classification problem. It can be solved either by using: machine learning algorithms or deep learning architectures.

Machine learning approach:

The problem can be solved either by using **logistic regression** or **support vector machine (SVM)**.

The problem with logistic regression is that it assumes a linear relationship between the dependent and independent variable, hence it performs badly on large datasets (like IMDB Reviews dataset) with too many features. Therefore it is better to use SVM with a non-linear kernel to solve the sentiment analysis problem.

Deep learning approach:

We can use three type of architectures to solve the problem:

1. Fully connected Neural Networks.
2. 1D Convolutional neural networks.
3. Recurrent Neural Networks (RNNs)

The problem with the two first solutions is that they don't take into account past inputs (which means each input is processed independently) because they don't have memory. Since we are dealing with a NLP problem, it is important to take in consideration past inputs to be able to highlight important words. LSTM layers used in RNN solve this issue, however they are hard to train and tend to overfit.

Models

SVM model:

For the machine learning approach we choose the SVM algorithm.

The model was trained using the "**rbf**" kernel, the hyper-parameter gamma was set to the inverse of the data variance (X.var()). The duration of training was **30mins.**

Bidirectional RNN:

For the deep learning approach, we built a bidirectional RNN with LSTM layer. Because of their ability to process data in chronological order and anti-chronological order, they are more likely to catch complex patterns than a simple RNN layer.

The training took **one hour.**

Performance

In our case the SVM model performed better than the RNN model.

The SVM model was **89%** accurate on the validation set, while the RNN model was **83%** accurate.

The SVM model was **98%** accurate on the **Test Dataset**. It correctly classified **12194** negative reviews as negative and misclassified **196** positive reviews. It correctly classified **12276** positive reviews as positive and misclassified **236** negative ones.

The RNN model suffers from overfitting which we tried to solve by increasing the probability of dropout, applying regularization techniques, changing the optimizer and activation functions, but still they didn't work.