Deep Learning for Sentiment Analysis

Authored by:
Abhijeet Sant(ars1125) and Manasi Pai(msp561)

Contents

Abstract:

With the advent of social media, opinions of general population matter a lot to analyze how the broadcast of information impacts the business of an organization. Decoding public sentiments is essential for many applications including review of a product, predicting political mood of a nation, success of recently launched movies, books or songs, or currently in-trend fashion. Review or opinion on large-scale networks like Twitter, Facebook, Instagram and other platforms are now central to humans and plays a great influencer in our behavior. Sentimental Analysis is study of people's sentiment, inkling, opinion using concept called as natural language processing to determine whether the polarity towards the subject is positive, negative or neutral in nature. After its launch tweeter has been one of the most popular social media platforms across the world. People easily share their opinions, reviews spontaneously and swiftly. Use of hash tags helps categorize series of tweets within the same topics. In this project we use rapid miner to determine the sentiment of users for popular airlines brand. The primary aim is to develop a deep learning model for analyzing sentiment score in noisy twitter streams. Results we obtains determine the accuracy of our model in correctly predicting the inclination of the tweets towards the three sentiments.

Introduction:

With an unprecedented growth in social media usage, platforms like Twitter, Facebook, Instagram, Tumblr, Pinterest and many more are now being used to share thoughts, reviews and opinions. Every minute approximately 350,000 tweets are sent containing varied set of expressions, many including personal experience and opinions on popular brands and other products. Millions of users express their actual experience about brands they interact with. Twitter has become the goldmine to analyze brand performance. Many companies are now using Twitter data to analyze public sentiments about their products to improvise and monetize their businesses.

In this project, we are building a deep learning model for sentiment analysis of Twitter content to analyze sentiment of customers for a popular American airline brand. Opinions and expressions on Twitter are raw, casual, candid and informative than what can be picked from formal surveys etc. These sentiments, if identified, can be useful for companies to not only monitor their brand's performance but also identify areas of improvement and other factors influencing polarity of these sentiments. These brands can be products, celebrities, events or political parties.

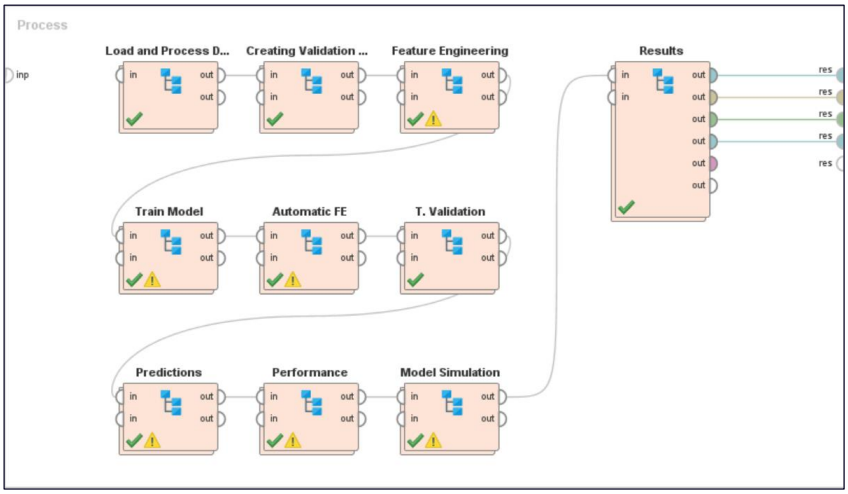
The system is evaluated on tweets regarding a popular airline brand dataset. The proposed model outperforms current baseline models (Naive Bayes, which shows that going beyond the content of a document (tweet) is beneficial in sentiment classification, because it provides the classifier with a

deep understanding of the task. Based on words (30 most important words in the tweet for this project) in the user review/tweet and assigns it to one of the following classes that best reflects its sentiment: positive, negative or neutral. We adopt a deep learning approach and use a combination of published proven features to predict the sentiments. A positive and negative class would contain polar tweets expressing a sentiment. However, a neutral class may contain an objective or subjective tweet either a user reflect neutrality in an opinion or contain no opinion at all. For e.g. consider below table.

Class	Statement	Key words
Positive	I enjoyed their services. The movie was good. He will win.	'enjoyed', 'good', 'win'.
Negative	I hate chocolate. The movie was extremely bad. He will never win.	'hate', 'bad', 'never'.
Neutral	There are 12 months in a year. Today is Tuesday. Mary had a little lamb.	-

Analysis:

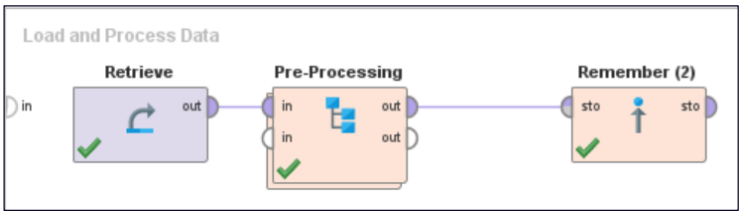
This project demands sentiment analysis of tweets. The below process has been adopted in order to achieve maximum accuracy:



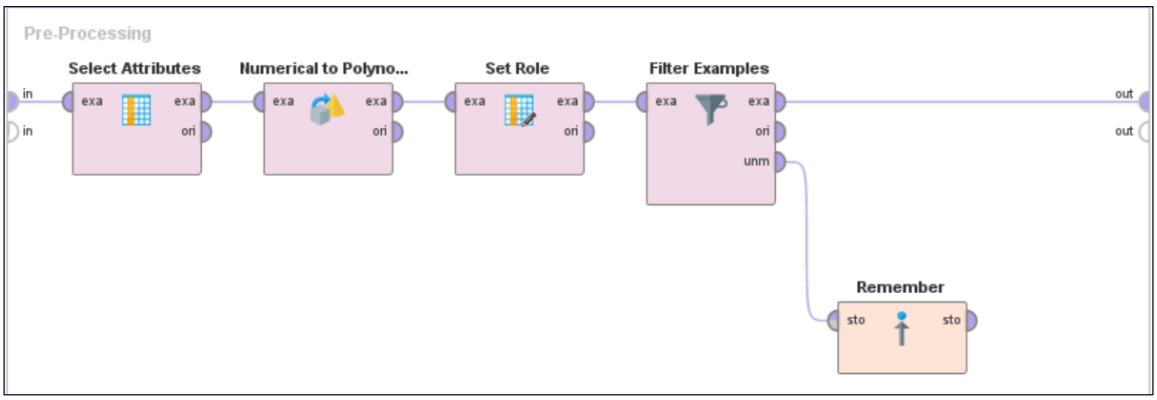
Let us deep dive into every sub-process with its function:

Sub-Process

1. Load and Process Data:



- In Retrieve Operation, we load our dataset into Rapid Miner using Import Data. We then carry out certain pre-processing operations on the data set before creating models for predictions.

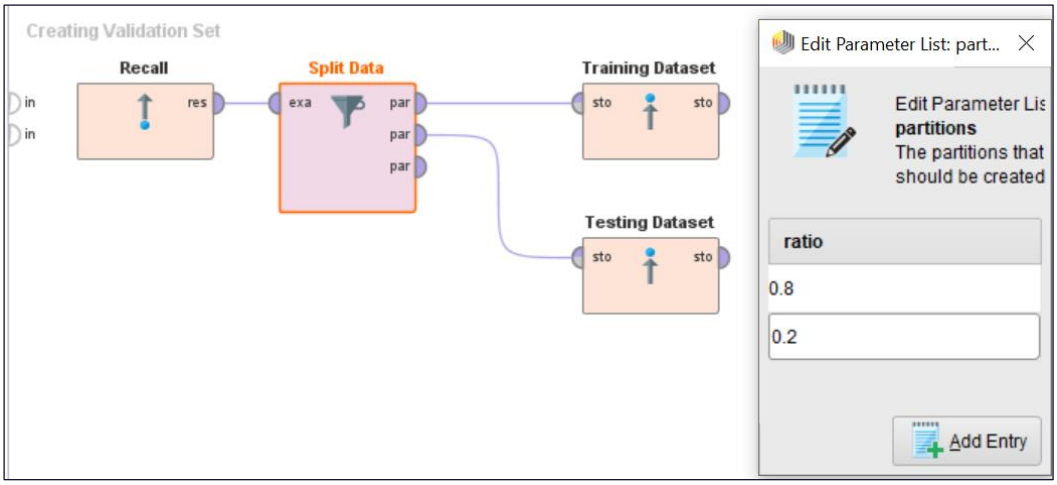


Selection of all Attributes having an impact on result of the model is one of the major steps carried out in this pre-process.

After selecting out features, we then convert sentiment attribute from Numerical to Polynomial and set the role (select target variable) to that feature we want to predict.

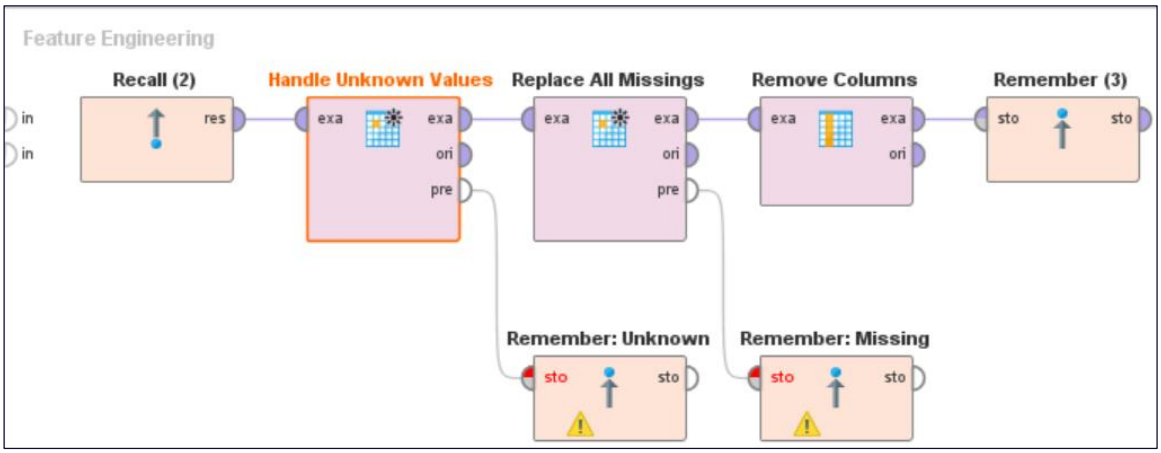
We filter out all the missing values using the filter examples operations. Having missing values in your data is not necessarily a setback but it is an opportunity to perform right feature engineering to guide the model to interpret the missing information right way.

2. Creating Validation Set:



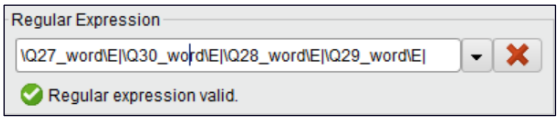
This sub-process has been created to partition the data into Training and Testing set. In Training set we train our model on various parameters so that the model can be well trained and accurately predict the sentiment in our testing data.

3. Feature Engineering:



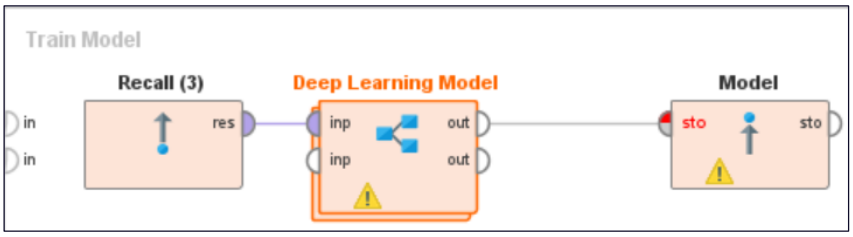
This sub-process will remove all the unknown and missing values from the dataset, if available along with some attributes which are not properly contributing to the predictions of sentiments.

Using a regular expression, we have eliminated last 4 columns for better accuracy. As some of the tweets do not contain more than 25 words, the value in last 4 columns was mostly zero. This impacted the accuracy of the overall model and hence we decided to eliminate the columns.

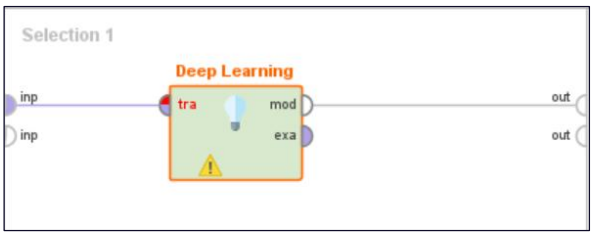


4. Train Model:

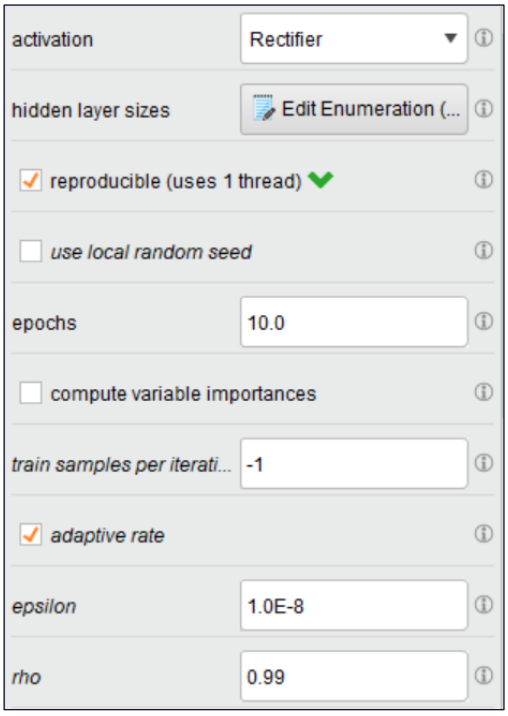
We then train our data set using the deep learning model.

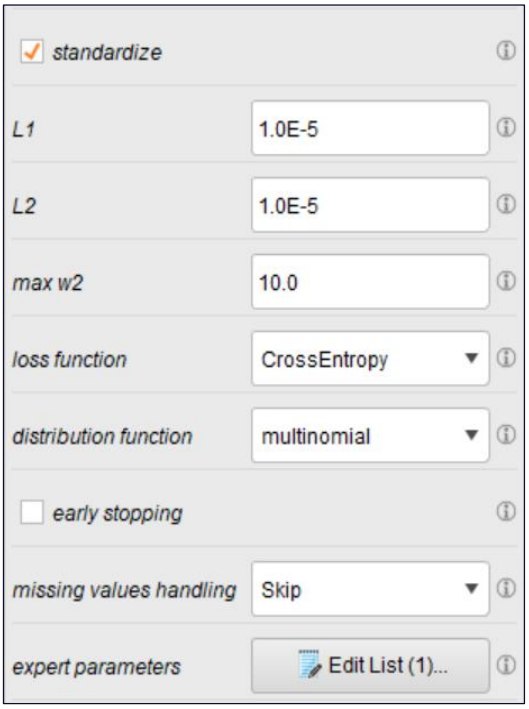


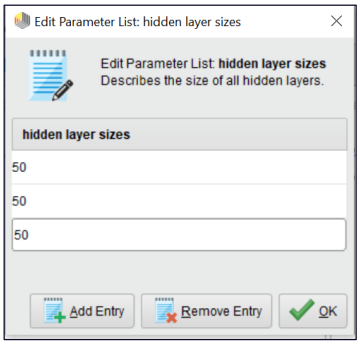
The Deep Learning Model sub-process contains the below operator:

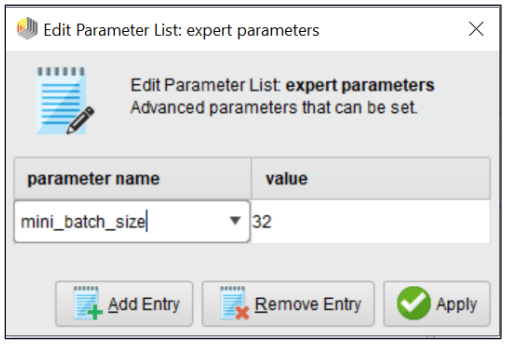


For better model performance, we have altered below parameters of our deep learning model:

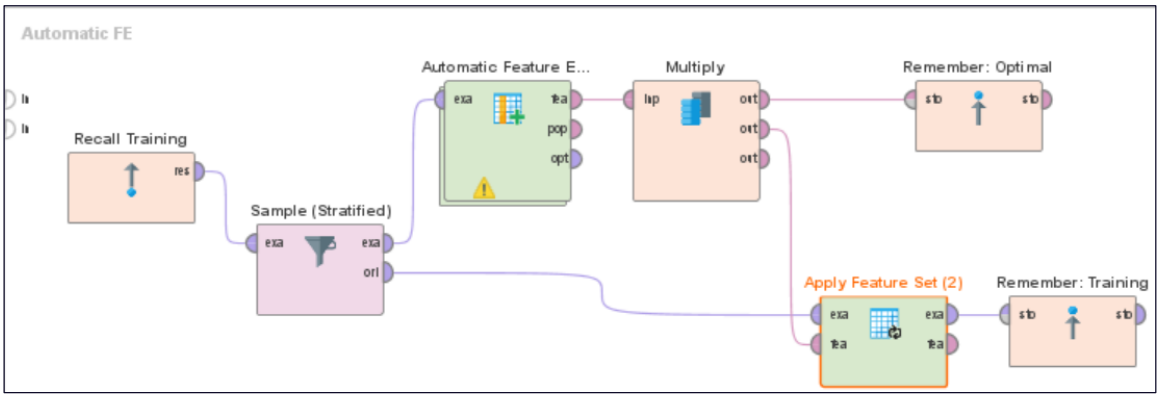


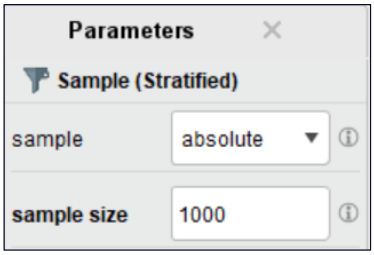




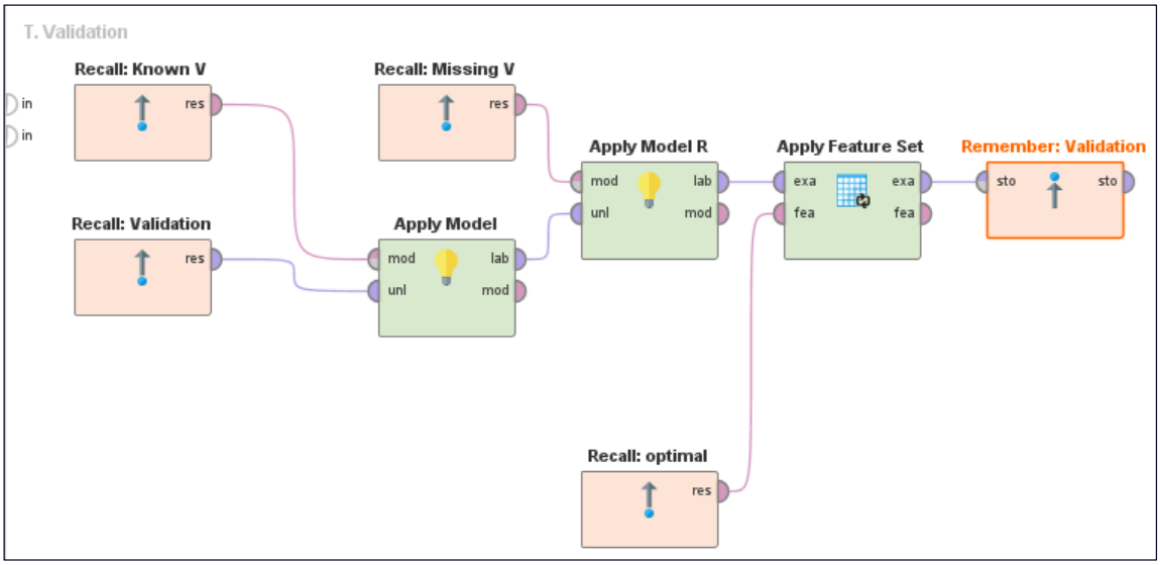


5. Automatic Feature Engineering:



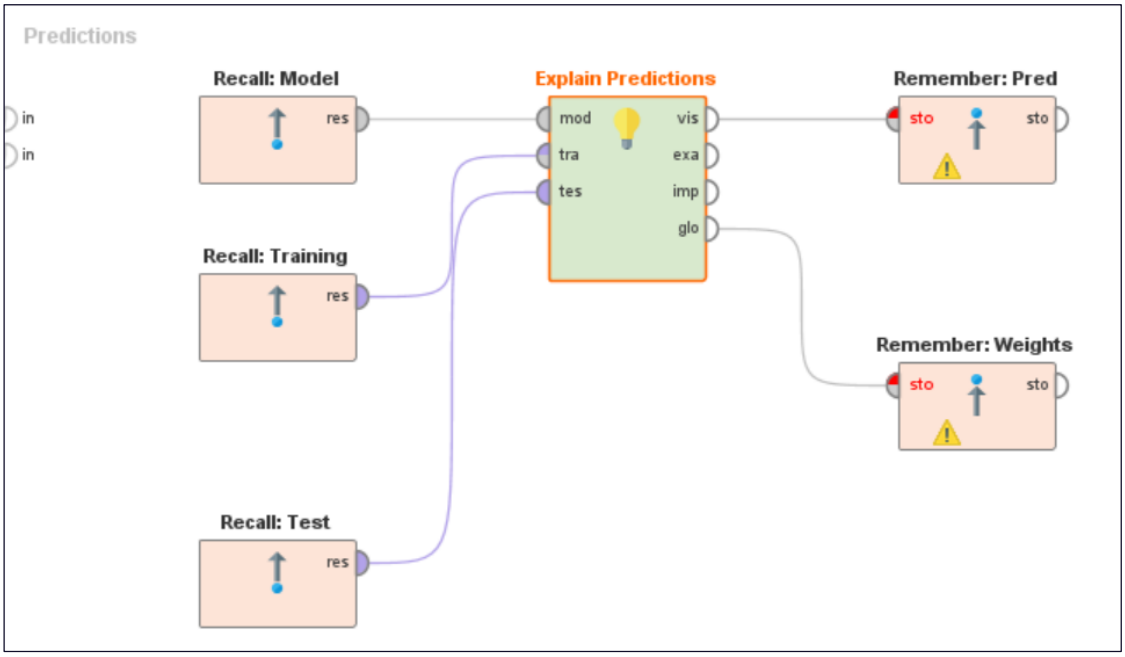


6. Test Validation:

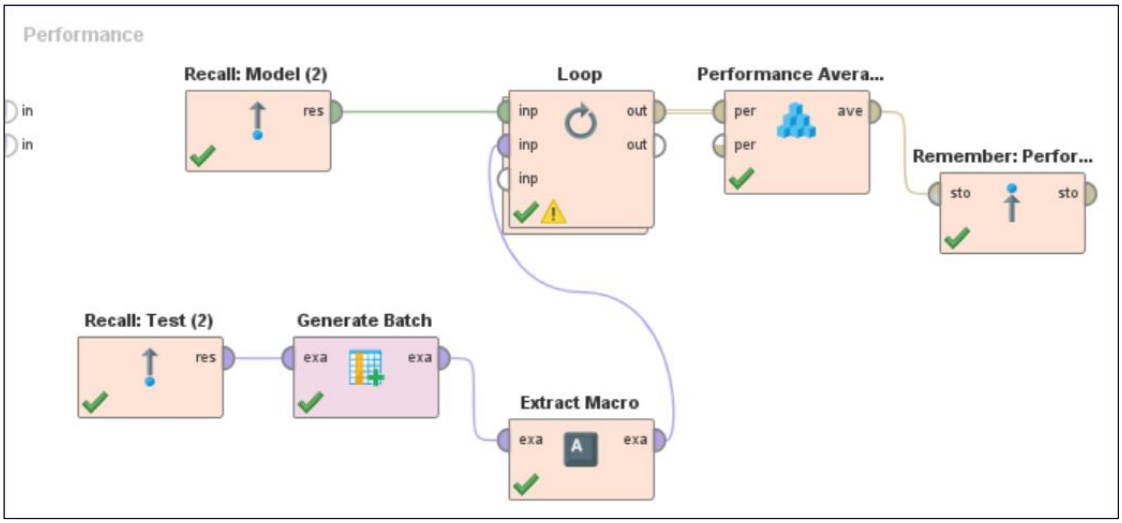


7. Predictions:

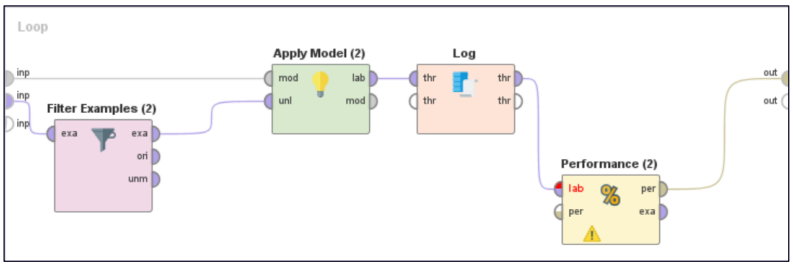
Once we have trained our model, we apply the trained model on our testing data. The model then predicts the sentiment of each instance of test data.



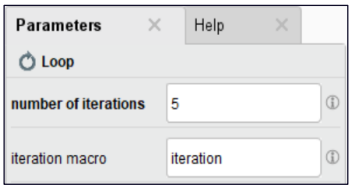
8. Performance:

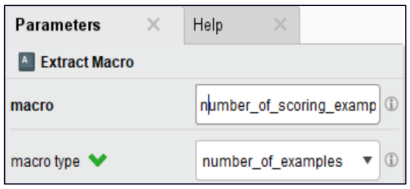


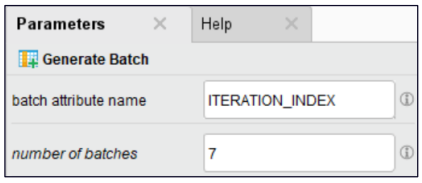
Loop:



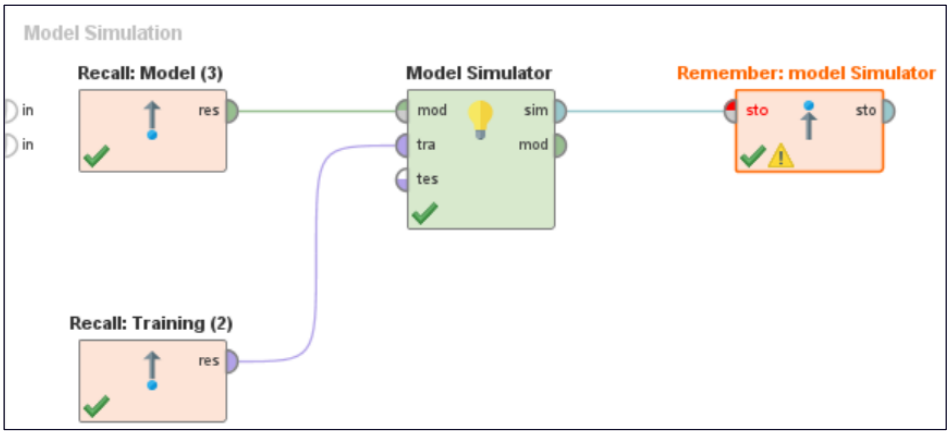
Parameter details of the operators in Loop:





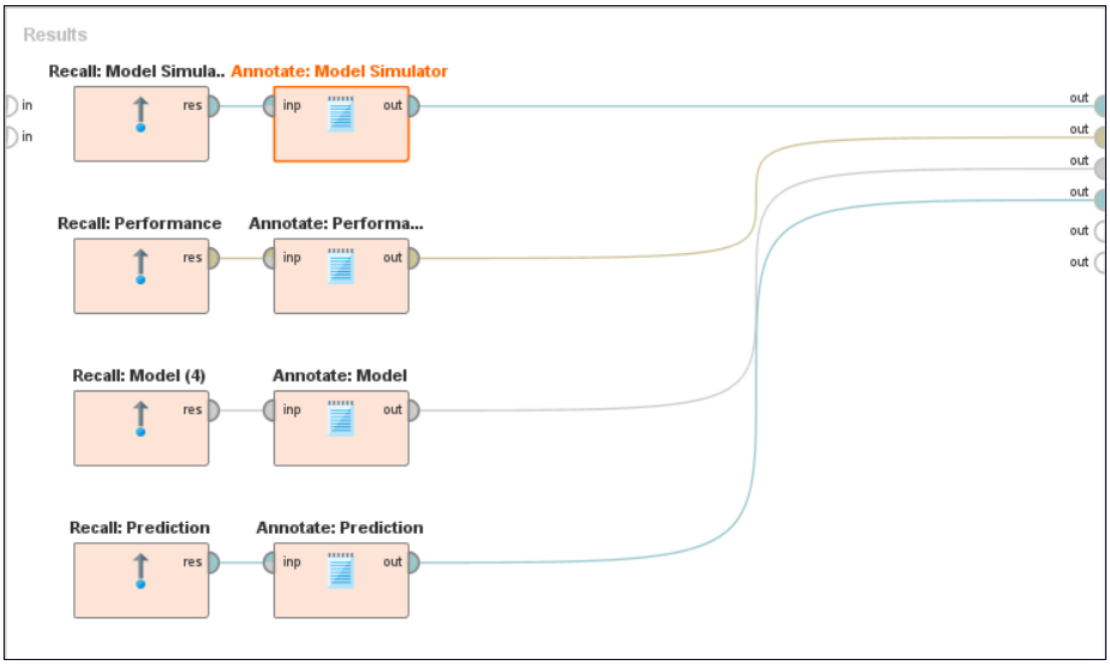


9. Model Simulator:

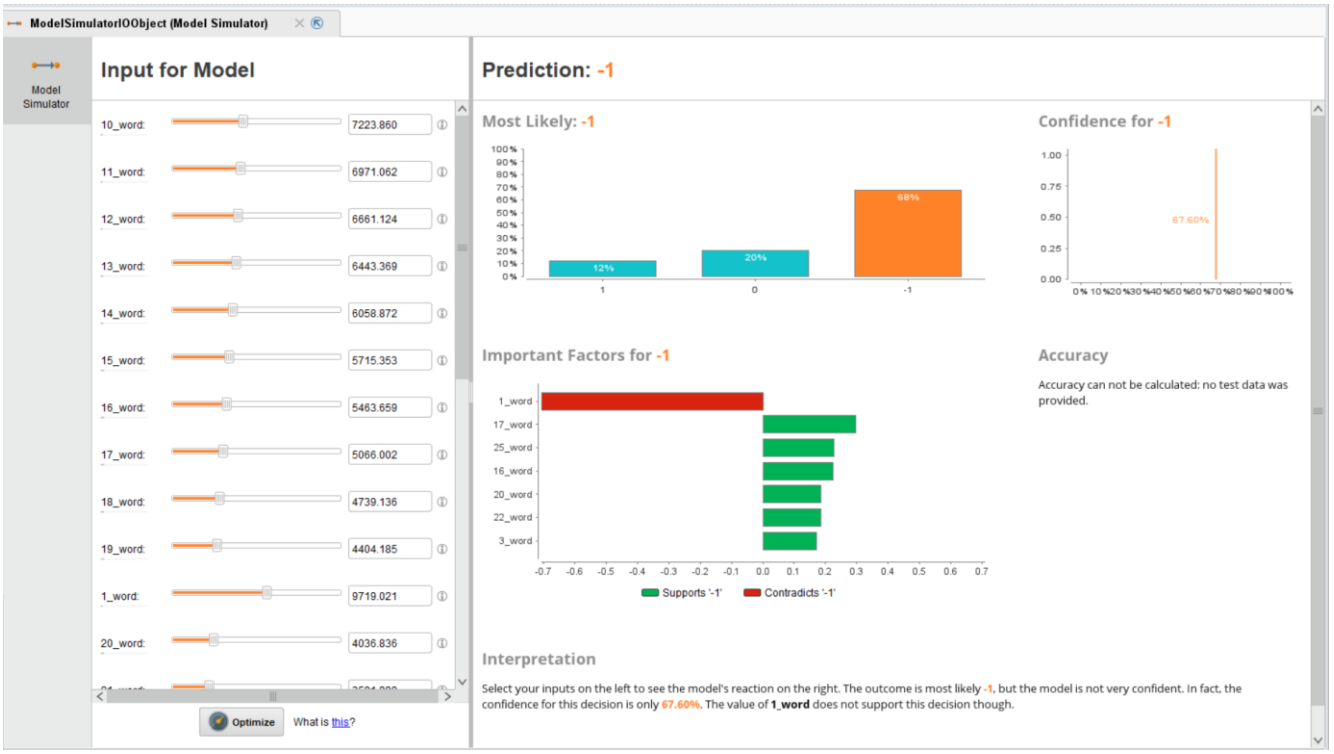


10. Results

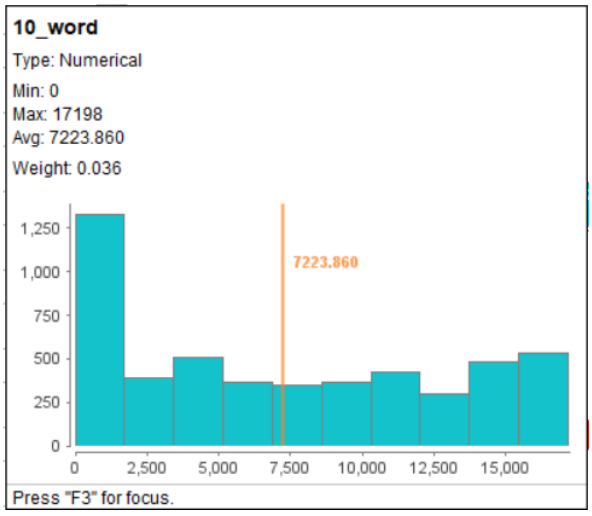
Following is the interior of result sub-process which comprises of Model Simulation, Performance, Model and Prediction Recalls obtained from the above processes.



a. Model Simulation:



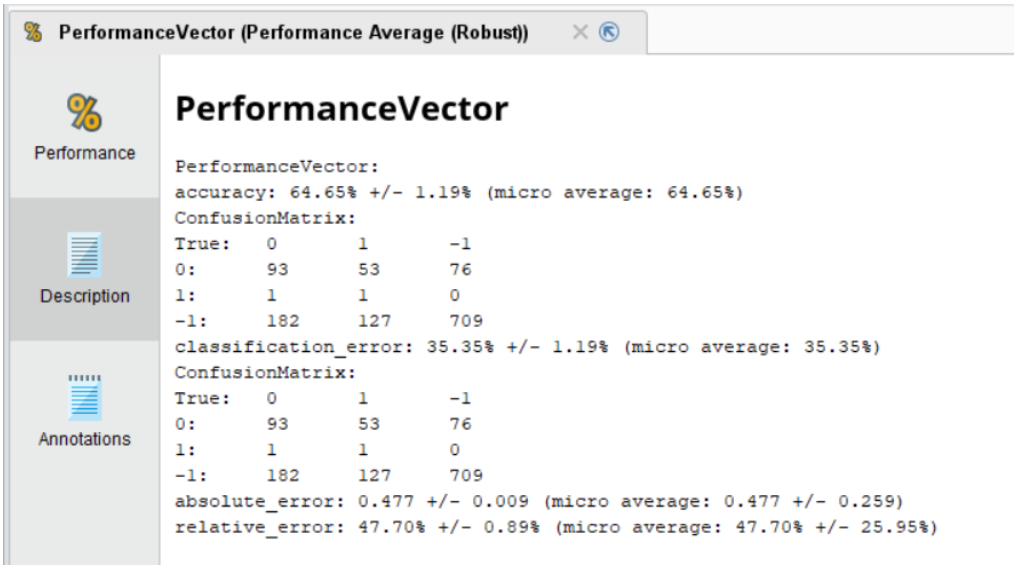
Weights for each word can be found by hovering around the i for each word. For example:



b. Performance:

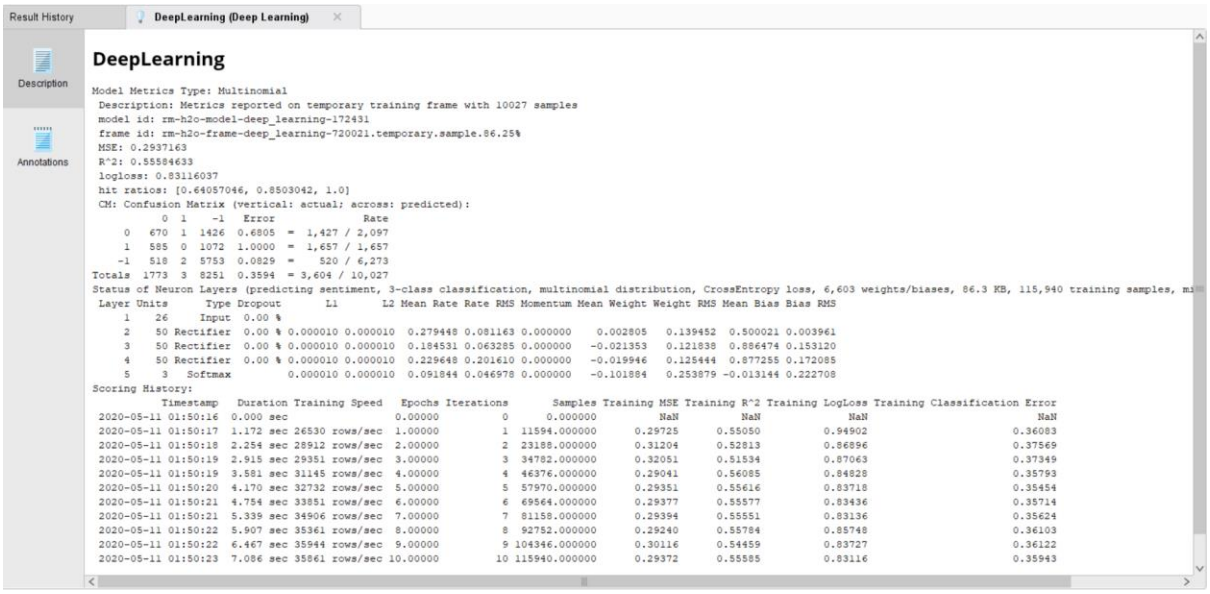
Performance vector helps us understand our overall performance of model in detail. We have achieved an accuracy of **64.65%.** It also gives detailed data of number of instances in each of the 3 sentiments.

We have also achieved, an absolute error as low as 0.44 which indicates that there is miniscule difference between the actual and predicted value.



c. Model:

The below image gives us detailed information about the deep learning model we used to train our data set.

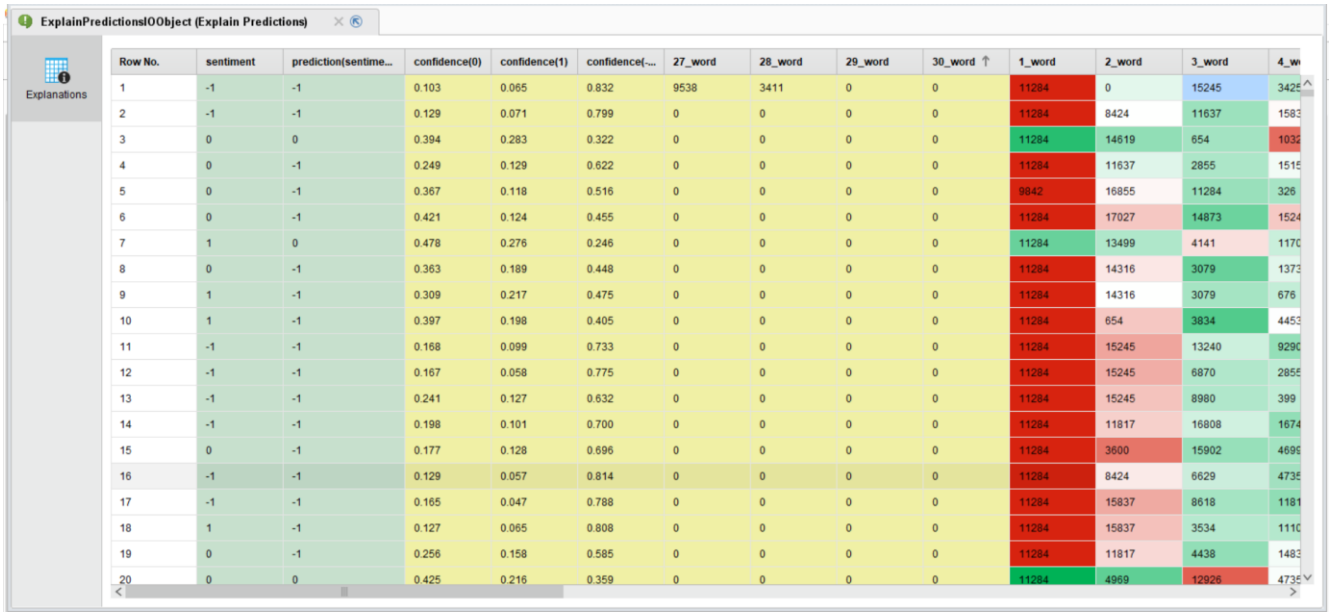


It contains information like Confusion Matrix which is a performance measurement to classify our predicted and actual values, MSE (Mean Squared error) is an estimator which helps us identify the error between our actual and predicted values, R^2.

It also gives detailed values of Neural Network Layers along with additional information like L1, L2, Mean Rate which is mean of learning rate, Number of Neurons in layer,

Activation function and Scoring history.

d. Predictions:



Row No.	sentiment	prediction(sentime...	confidence(0)	confidence(1)	confidence(-...	27_word	28_word	29_word	30_word	1_word	2_word	3_word	4_w
1	-1	-1	0.103	0.065	0.832	9538	3411	0	0	11284	0	15245	3425
2	-1	-1	0.129	0.071	0.799	0	0	0	0	11284	8424	11637	1583
3	0	0	0.394	0.283	0.322	0	0	0	0	11284	14619	654	1032
4	0	-1	0.249	0.129	0.622	0	0	0	0	11284	11637	2855	1515
5	0	-1	0.367	0.118	0.516	0	0	0	0	9842	16855	11284	326
6	0	-1	0.421	0.124	0.455	0	0	0	0	11284	17027	14873	1524
7	1	0	0.478	0.276	0.246	0	0	0	0	11284	13499	4141	1170
8	0	-1	0.363	0.189	0.448	0	0	0	0	11284	14316	3079	1373
9	1	-1	0.309	0.217	0.475	0	0	0	0	11284	14316	3079	676
10	1	-1	0.397	0.198	0.405	0	0	0	0	11284	654	3834	4453
11	-1	-1	0.168	0.099	0.733	0	0	0	0	11284	15245	13240	9290
12	-1	-1	0.167	0.058	0.775	0	0	0	0	11284	15245	6870	2855
13	-1	-1	0.241	0.127	0.632	0	0	0	0	11284	15245	8980	399
14	-1	-1	0.198	0.101	0.700	0	0	0	0	11284	11817	16808	1674
15	0	-1	0.177	0.128	0.696	0	0	0	0	11284	3600	15902	4696
16	-1	-1	0.129	0.057	0.814	0	0	0	0	11284	8424	6629	4735
17	-1	-1	0.165	0.047	0.788	0	0	0	0	11284	15837	8618	1181
18	1	-1	0.127	0.065	0.808	0	0	0	0	11284	15837	3534	1110
19	0	-1	0.256	0.158	0.585	0	0	0	0	11284	11817	4438	1483
20	0	0	0.425	0.216	0.359	0	0	0	0	11284	4969	12926	4735

Conclusion:

Sentiment analysis in the domain of social platforms is a relatively new research topic. A lot of challenges are involved in terms of tone, words used, language, geography and grammar of the tweets. Nowadays, sentiment analysis or opinion mining is a hot topic in various aspects of data fields. Twitter is a source of vast unstructured and noisy data sets that can be processed to locate interesting patterns and trends.

Accurate sentiment detection of corpus of texts is still very obscure because of the complexity in the English language. This complexity further increases when we consider data from other languages as twitter supports a total of 61 languages. In this project we tried to show the basic way of classifying tweets into positive, negative or neutral category using Naive Bayes as baseline model (55%) and deep learning model.

Note: We tried to proficient the baseline performance model and hence the accuracy obtained from Naïve bayes is high. But the obtained baseline nascent accuracy of Naïve Bayes was around 55%.