SENTIMENTAL ANALYSIS USING TWITTER

Final Project

ISTE.470.600

Dr. Khalil Al Hussaeni

Report By:
Adit Dhall
Hughann Plucena
Jaiveer Kapadia
Sandeep Mamidala

Manifesto

We believe that everyone in the group contributed equally to this project. While doing the practical work, we were all together on Zoom, but it was done on Adit’s PC since he was already registered for the educational account. As for the report, we had divided the work amongst ourselves.

Table of Contents

Table Of Figure

Introduction

As a group, we had decided to create a Sentiment Analysis of one of the biggest social media platforms, namely Twitter. We aimed to extract data on specific hashtags and analyze whether the tweets gathered were positive, negative, or neutral tweets in terms of sentiment.

The controversy that we decided to tackle was the problem that surrounds the Indian government's decision to pass the bill that the farmers in the country deemed to be 'anti-farmer law.' The more specific issue we're talking about is two of the more well-known actors in India: Diljit Dosanjh and Kangana Ranaut. Diljit and Kangana caused an explosion on Twitter by publicly arguing back and forth on the social media platform. Diljit favors the protest, whereas Kangana is against the protest. We chose this topic in order to gather the information/demographic of both actors and who seem to have more supporters. The result of this could be used to find out how many people are reacting to this issue.

This problem has not been seen nor solved before since this one is a more recent problem. The main challenge of this case study was to determine the accuracy of the result. We've utilized a couple of different APIs' to see which one yields better results. We ultimately used an API called Vader. Vader allowed us to gather a larger dataset than the other APIs that we tried to use so this was a plus as well.

Approach

Overview

We used Vader (Valence aware dictionary for sentiment reasoning) API, which is a pre-trained model. This helped us get the sentiment score that we fetched from Twitter.

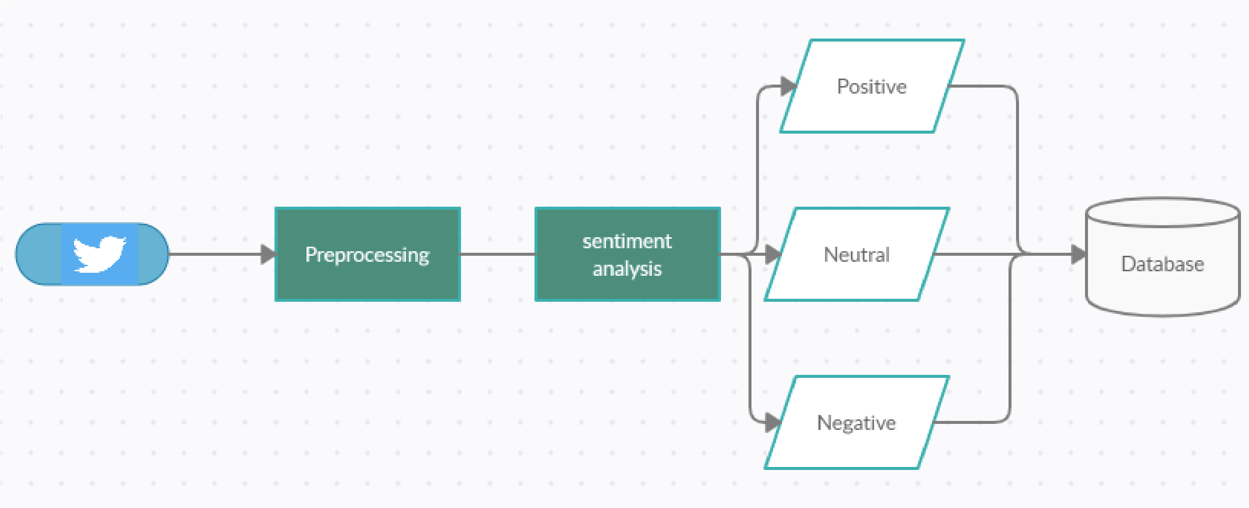


Image 1:Components for our approach

Dataset

As our topic suggests, we got our dataset from the social media platform, Twitter. Every time we run the process on RapidMiner, our datasets get populated with the latest tweets. Our dataset contains tweets featuring Diljit and Kangana, who are arguing with each other over the farmer's protest which is going on in India, and also tweets in which they are tagged by others, who are either with them or against them. Each record will represent a tweet from a user, an ID, a row number, sentiment score, and a sentiment. There are two data objects in our dataset: Twitter 1 and Twitter 2, and we tried the analysis with a total of 1029 records in our dataset, but this can be expanded by changing the search criteria in the twitter operator. There is a total of 5 attributes:

	Row No.	Id	Text	Score	Sentiment
Data Type	Autoincrement	BigInt	Text	float	Text
Domain Value	0-99999	1 to 20000000000000000000	-	-10.0 to 10.0	-
Description	Row no. of each record	Unique ID assigned to every tweet	This is the tweet by the user	Score given to the tweet (-10 to -0.05 = Negative, -0.05 to 0.05 = Neutral, 0.05 to 10 = Positive	Sentiment (Negative, Neutral, or Positive) given to each tweet.

Table: Dataset Information

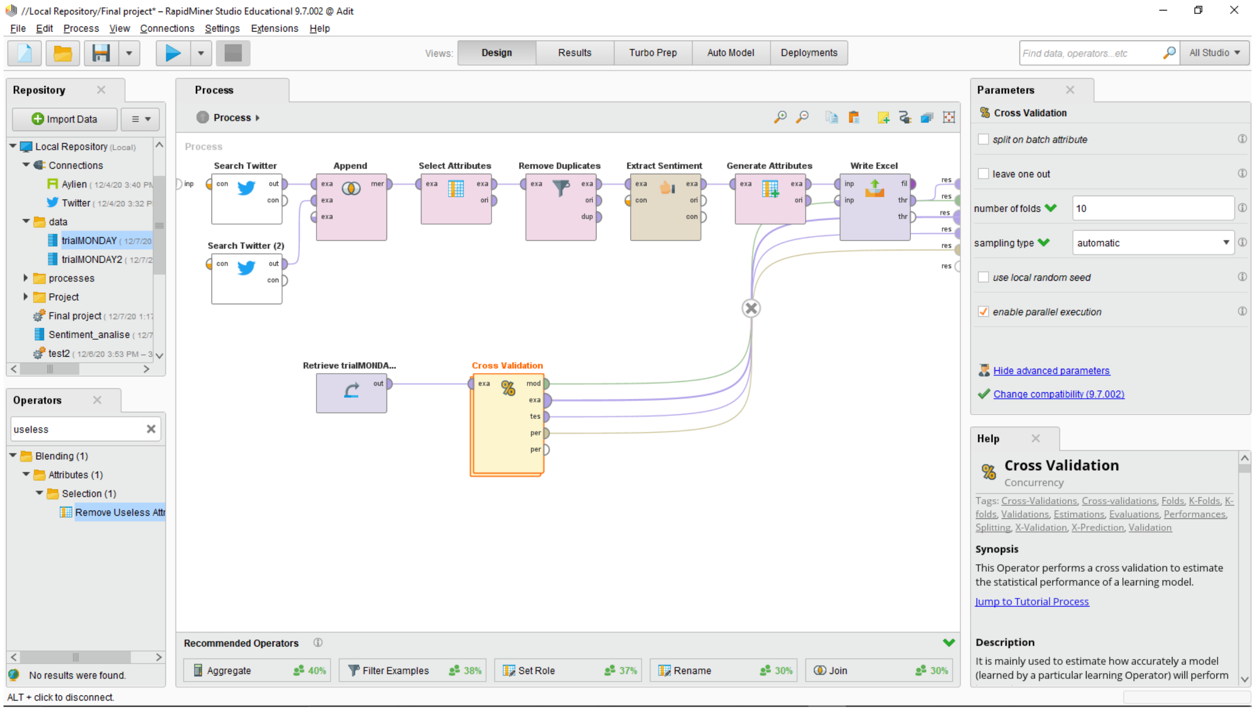
Pre-Processing

Pre-processing is one of the essential steps for data mining. We had to analyze our data thoroughly. There would’ve been many complications in our dataset, such as missing values, inconsistent values, duplicate values, etc. For the preprocessing, we used data reduction and data integration. We fetched datasets from two twitter objects and then integrated them into one dataset using data integration and then finally searched for duplicate tweets using data reduction technique.

Experimental Analysis

We created two Twitter operators; both were extracting different data with different queries. The data collected was appended so that they were in the same dataset. Then the attribute "Text" was selected and was searched for any duplicate entries; if there were, they were deleted. The next step was to analyze the tweets' sentiment. For this, we used an extract sentiment operator, and the model used is called Vader. As Vader only provides a sentiment score, we used the "Generate Attributes" operator, which was used to create another attribute named "Sentiment." The conditions used for this were: If the score is between -10 to -0.05, then sentiment would be Negative, else if the score is between -0.05 to 0.05, then sentiment would be Neutral else if the score is between 0.05

to 10, then sentiment would be Positive. After this, the results were saved to an excel file.



Cross Validation

- ☐ split on batch attribute
- ☐ leave one out
- number of folds: 10
- sampling type: automatic
- ☐ use local random seed
- ☑ enable parallel execution

Hide advanced parameters

Change compatibility (9.7.002)

Cross Validation

Concurrency

Tags: Cross-Validations, Cross-validations, Folds, K-Folds, K-folds, Validations, Estimations, Evaluations, Performances, Splitting, X-Validation, X-Prediction, Validation

Synopsis

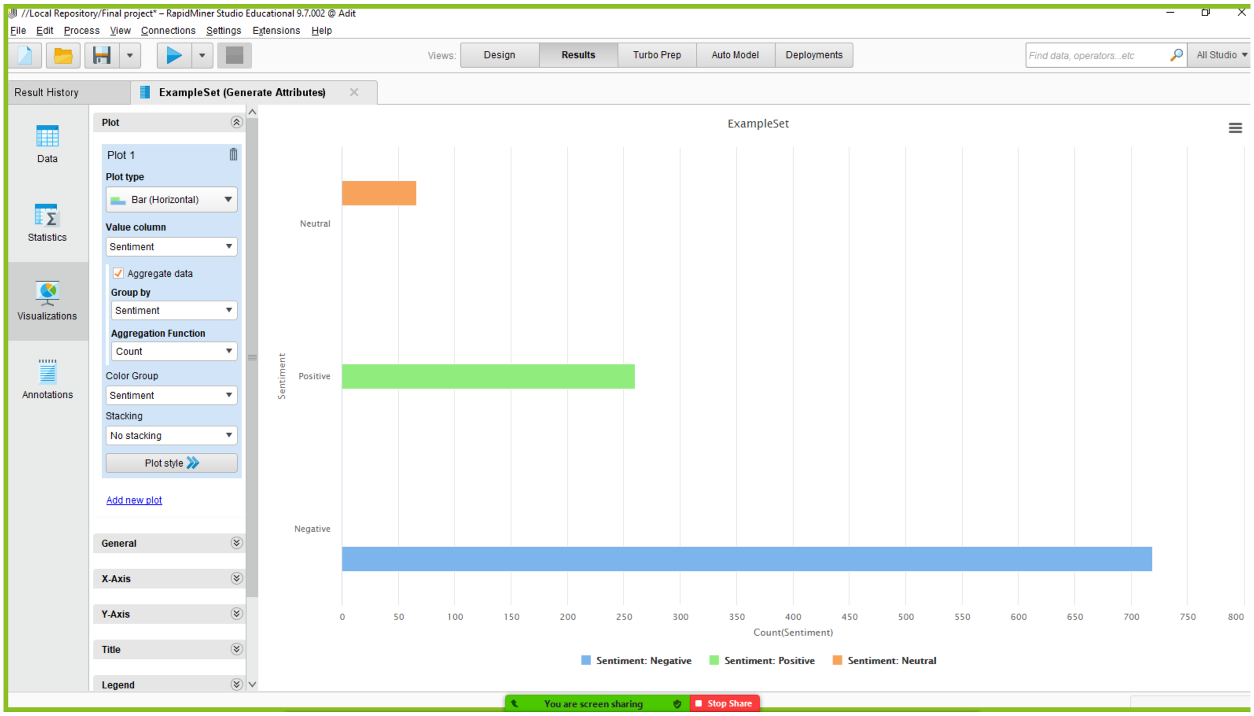
This Operator performs a cross validation to estimate the statistical performance of a learning model.

Jump to Tutorial Process

Description

It is mainly used to estimate how accurately a model (learned by a particular learning Operator) will perform

Image 2: Main Process



//Local Repository/Final project* – RapidMiner Studio Educational 9.7.002 @ Adit

ExampleSet (Generate Attributes)

Plot 1

Plot type: Bar (Horizontal)

Value column: Sentiment

☑ Aggregate data

Group by: Sentiment

Aggregation Function: Count

Color Group: Sentiment

Stacking: No stacking

Image 3: Sentiment Analysis of Tweets

For performance metrics, we checked our program's efficiency by running the process repeatedly and increasing the number of tweets in each attempt. Below is the table with the number of tweets analyzed with their runtimes:

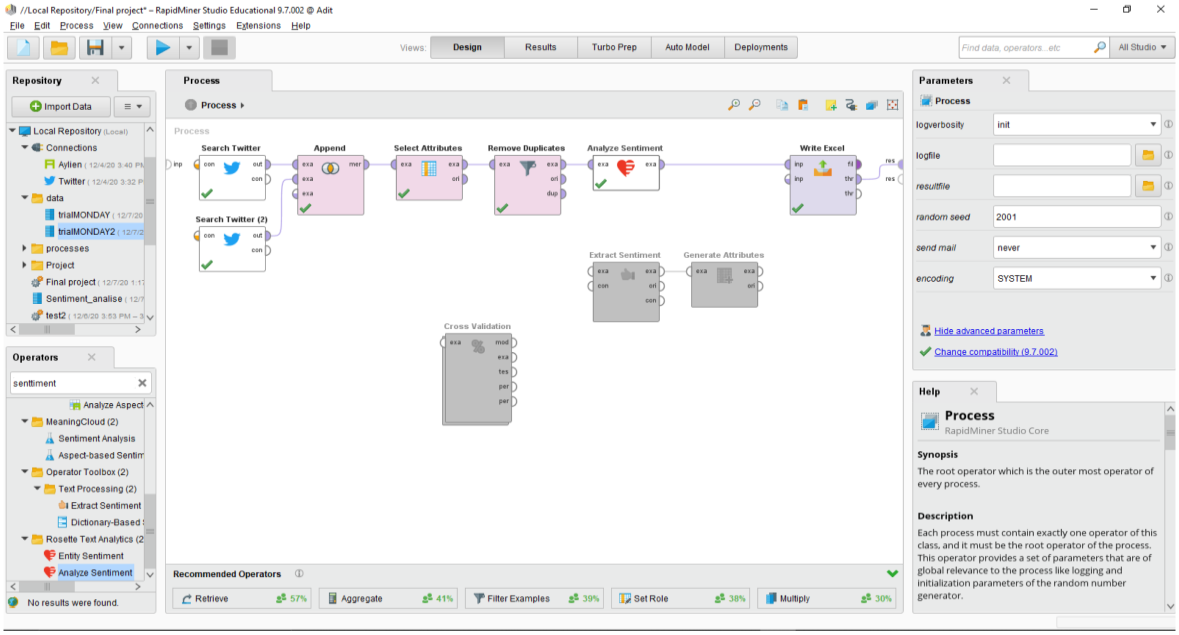
Attempt No.	No. of Tweets	Runtime
1.	2000	10.2 sec
2.	4000	19.4 sec
3.	4500	23.9

Table: Process Runtime

The main time was taken by the “search Twitter” operator to fetch tweets. It took approx. 4.9 sec/1000 tweets. The main API that we used “Vader,” took less than 1 sec.

We believe that our model ran efficiently and as per our expectations since Vader’s performance did not slow down as much and the difference would be negligible even though we increased the number of tweets that was supposed to be analyzed.
From the results, we can see that people are taking sides and arguing with each other and this is why the graph shows that the overall sentiment is negative. This kind of information can help the government make decisions based on these results. For example, if they see that the public is starting to support the farmer's movement, they may impose strict rules to ensure that things don't go out of control. And this is just one example of how this information can help. Another one would be for the social media marketing campaigns for the actor or actress to make quick decisions such as whether they should continue tweeting about the issue depending on how it would affect their reputation.

Initially, we had used an API/plugin called Rosette text analysis, which we implemented to help produce the sentiment analysis results, but the problem with this was that it was not accurate and had limited attempts.



logverbosity: init
logfile:
resultfile:
random seed: 2001
send mail: never
encoding: SYSTEM

Process
RapidMiner Studio Core

Synopsis
The root operator which is the outer most operator of every process.

Description
Each process must contain exactly one operator of this class, and it must be the root operator of the process. This operator provides a set of parameters that are of global relevance to the process like logging and initialization parameters of the random number generator.

Image 4: Analysing Tweets Using Rosette API

After some research, we found another API called MonkeyLearn, which we read was efficient and accurate but did not hold up to those claims in our implementation. And we were getting similar results as with Rosette. We decided to do some more research, which ended up with us finding another API called Vader. For checking the accuracy, we selected tweets manually and read them to see if the sentiment status given to them was correct. We were thrilled with Vader as the accuracy percentage was greatly increased as we randomly selected about 48 tweets and predicted the accuracy according to how we felt about the tweets and Vader was much more accurate as compared to the other 2 apis.

Conclusion

From the results, we can see that most of the general public in the sample of collected data were choosing sides between both parties. An ethical issue that we found with our project was the ability to take anyone’s tweet for analysis without needing their direct consent. Thus, it would be easy to analyze a specific ID’s/user’s tweets and determine what kind of person they are. This information can be beneficial for online marketing campaigns.

References

Monali, Yadav, and Shivani Raskar. "Sentimental Analysis On Audio And Video Using Vader Algorithm". *Irjet.Net*, 2020, https://www.irjet.net/archives/V6/i6/IRJET-V6I641.pdf. Accessed 7 Dec 2020.

"Rosette Text Analytics Extension For Rapidminer Predictive Analytics". *Rosette Text Analytics*, 2020, https://www.rosette.com/rapidminer/.

"Text Analysis API Documentation". *Docs.Aylien.Com*, 2020, https://docs.aylien.com/textapi/rapidminer-extension/#installation. Accessed 2 Dec 2020.