ELG5901

MVP sentiment Analysis

Uottawa supporter:

Prof. Dr. Murat Simsek

Egypt Mentor:

Dr. Mayada Hadhoud

Microsoft supporter:

Eng. Michel Naim

Student Name:

Dina Abdelhady

Table of contents:

Abstract:

This report aims to produce sentiment analysis classification predictions and compare them; analyze the pros and cons of algorithms and generate and communicate the insights and discuss the implementation steps of applying (building the model, transformation and evaluation, etc.) the strategy of the MVP was built on three of multiclass classification algorithms (Support Vector Machine and Naïve Bayes, Logistic Regression) then building Ensemble model. I will introduce the detailed steps of implementation of this strategy and discuss the achieved results.

Introduction:

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP).

Sentiment or opinion analysis is the use of natural language processing , computational linguistics and textual analysis in order to reveal the positive, negative or neutral feelings of a text towards the text's subject- Wikipedia.

The target of our project is knowing crowd perspective. So, Sentiment analysis is an import part.

Methodology:

1. For implementation, I used (Sentiment-140 data set from Kaggle) which contains two classes (0: Negative, 4 : positive). Using Textblob, I succeed to relabel the data to be (positive, negative and neutral) class.
2. After the preprocessing step, I split the data into training and testing data. Then use the under-sample technique to balance the training data.
3. For feature extraction, I used (TF-IDF, Bi-gram, Tri-gram and LDA).
4. Then, I trained and tested the SVM, NB and LR algorithms. Then I did an Error-Analysis for the Best models to get the miss-classified cases to correct it. Finally, I built the Ensemble model and save it.

The Dataset:

This is the sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api . The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

Implementation:

This strategy implemented using Python programming language. And some libraries such as :

- Scikit-Learn
- Numpy and Pandas
- Matplotlib
- Pickle

Result & Analysis:

Sentiment-140 dataset:

Let's take a look on Sentiment-140 dataset.

	text	target
0	@switchfoot http://twitpic.com/2y1zl - Awww, t...	0
1	is upset that he can't update his Facebook by ...	0
2	@Kenichan I dived many times for the ball. Man...	0
3	my whole body feels itchy and like its on fire	0
4	@nationwideclass no, it's not behaving at all....	0
...	...	...
1599995	Just woke up. Having no school is the best fee...	4
1599996	TheWDB.com - Very cool to hear old Walt interv...	4
1599997	Are you ready for your MoJo Makeover? Ask me f...	4
1599998	Happy 38th Birthday to my boo of alll time!!! ...	4
1599999	happy #charitytuesday @theNSPCC @SparksCharity...	4

1600000 rows × 2 columns

```
df['target'].value_counts()

4    800000
0    800000
Name: target, dtype: int64
```

I used only the 'text' and drop the 'target'

Cleaning :

text	clean_text
@switchfoot http://twitpic.com/2y1zl - Awww, t...	zl awww thats bummer shoulda got david carr th...
is upset that he can't update his Facebook by ...	upset cant update facebook texting might cry r...
@Kenichan I dived many times for the ball. Man...	dived many times ball managed save rest go bounds
my whole body feels itchy and like its on fire	whole body feels itchy like fire
@nationwideclass no, it's not behaving at all....	behaving im mad cant see
@Kwesidei not the whole crew	whole crew
Need a hug	need hug
@LOLTrish hey long time no see! Yes.. Rains a...	hey long time see yes rains bit bit lol im fin...
@Tatiana_K nope they didn't have it	nope didnt

After the cleaning, the data is ready to relabeled using the textblob

Textblob :

Textblob is a pre-trained model contains two sentiment analysis implementations, one of them is NaiveBayesAnalyzer (an NLTK classifier trained on a movie reviews corpus).

Here, I added a new column to my data frame has the new label for each tweet

text	clean_text	class
@switchfoot http://twitpic.com/2y1zl - Awww, t...	zl awww thats bummer shoulda got david carr th...	positive
is upset that he can't update his Facebook by ...	upset cant update facebook texting might cry r...	neutral
@Kenichan I dived many times for the ball. Man...	dived many times ball managed save rest go bounds	positive
my whole body feels itchy and like its on fire	whole body feels itchy like fire	positive
@nationwideclass no, it's not behaving at all....	behaving im mad cant see	negative

```
neutral     625144
positive    623317
negative    339644
Name: class, dtype: int64
```

Split the data :

```
xtrain :  1270484  xtest :  317621
```

```
y_train.value_counts()
```

```
neutral     499932
positive    498871
negative    271681
Name: class, dtype: int64
```

```
y_test.value_counts()
```

```
neutral     125212
positive    124446
negative     67963
Name: class, dtype: int64
```

Balance the Training data :

- **Under sample the biggest dataset:**

```
negative    271681
neutral     271681
positive    271681
Name: class, dtype: int64
```

<u>Feature engineering:</u>

<u>TF-IDF</u>

In order to re-weight the count features into floating point values suitable for usage by a classifier it is very common to use the tf–idf transform.

Tf means **term-frequency** while tf–idf means term-frequency times **inverse document-frequency.**

<u>For training data , **I** have 815043 tweets with 276218 words</u>

<u>For test data , **I** have 317621 tweets transformed on the same numbers of words.</u>

```
the size of X_train :   (815043, 276218)
the size of X_test :   (317621, 276218)
```

<u>N-gram</u>

- <u>bi-gram</u>

```
the size of X_train :   (815043, 2766644)
the size of X test :   (317621, 2766644)
```

- <u>tri-gram</u>

```
the size of X_train :   (815043, 6168783)
the size of X_test :   (317621, 6168783)
```

<u>LDA</u>

```
the size of X_train :   (815043, 3)
the size of X_test :   (317621, 3)
```

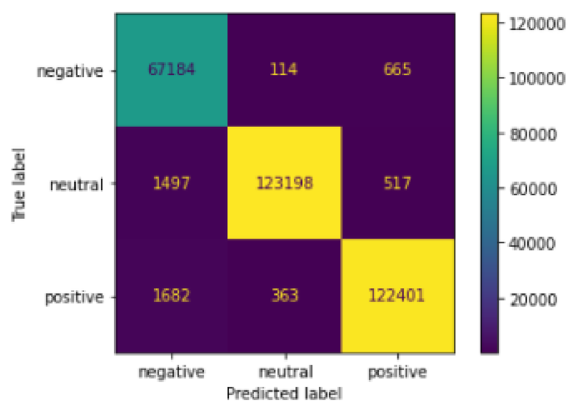
Build model:

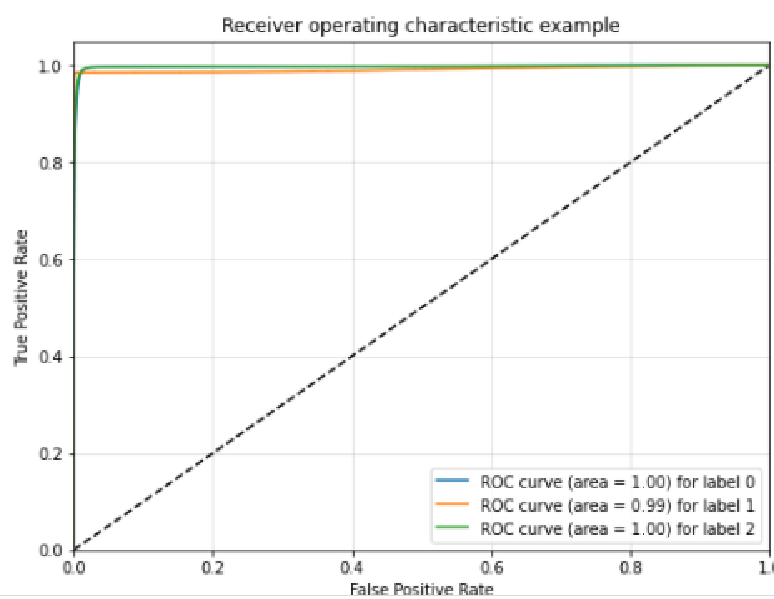
- **SVM**

With TF-IDF

the weighted f1_score: 0.9848342226675584

	precision	recall	f1-score	support
negative	0.95	0.99	0.97	67963
neutral	1.00	0.98	0.99	125212
positive	0.99	0.98	0.99	124446
accuracy			0.98	317621
macro avg	0.98	0.99	0.98	317621
weighted avg	0.99	0.98	0.98	317621



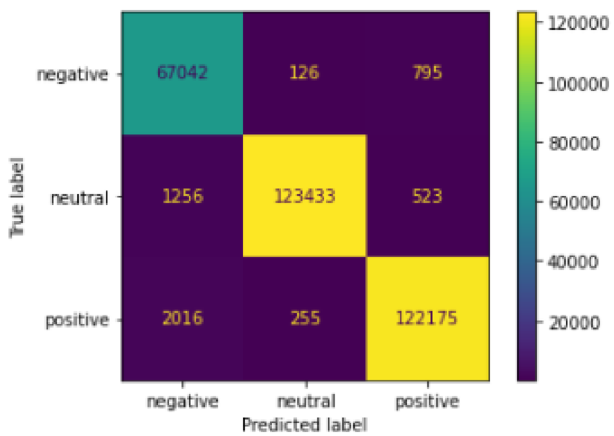


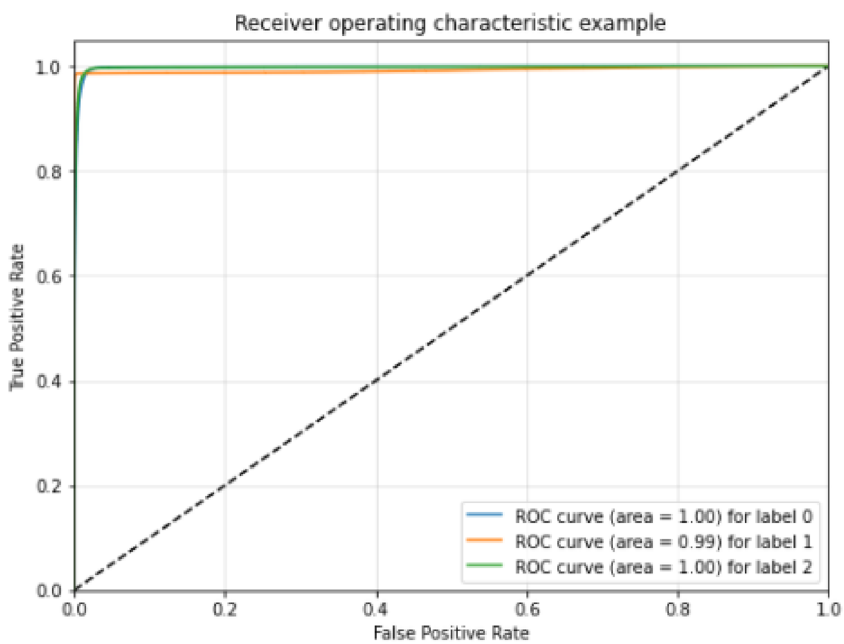
With bi-gram

```
the weighted f1_score: 0.9844206914331357
              precision    recall  f1-score   support

    negative       0.95      0.99      0.97     67963
     neutral       1.00      0.99      0.99    125212
    positive       0.99      0.98      0.99    124446

    accuracy                           0.98    317621
   macro avg       0.98      0.98      0.98    317621
weighted avg       0.98      0.98      0.98    317621
```



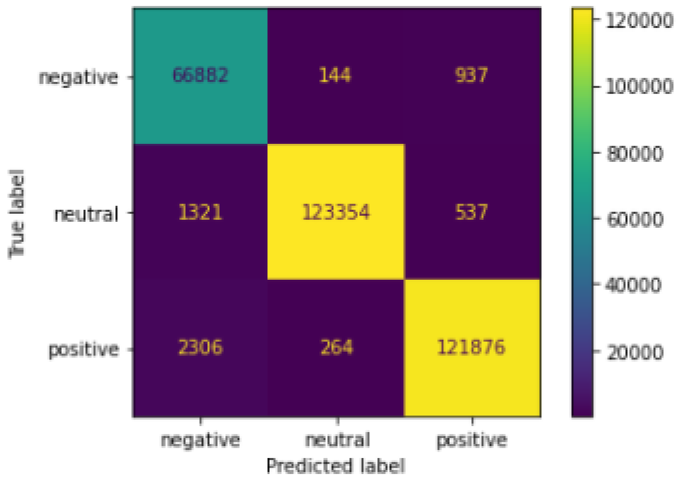


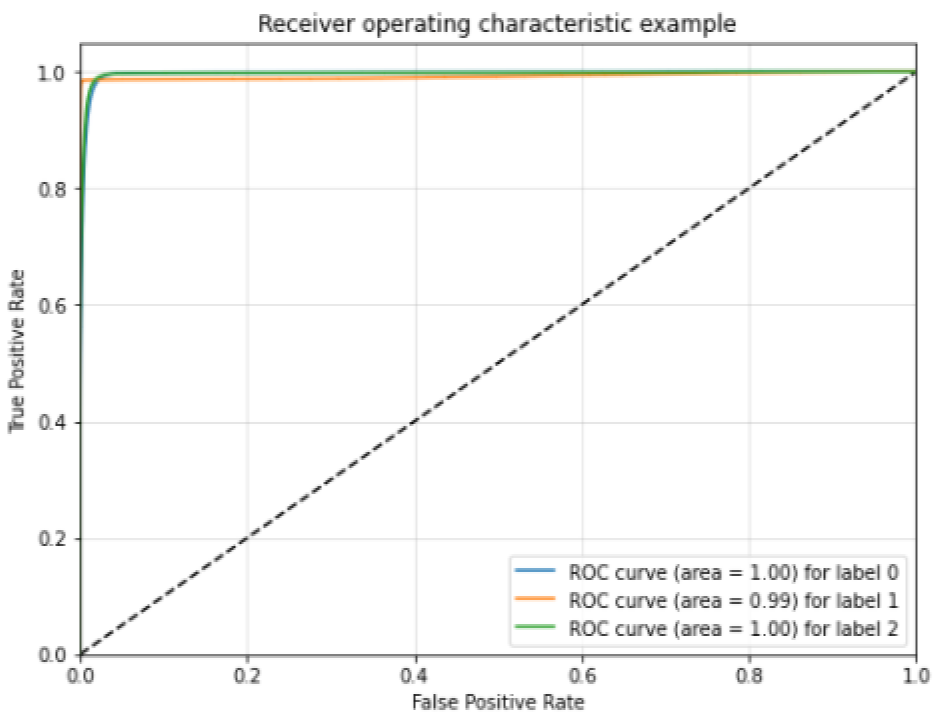
With tri-gram

```
the weighted f1_score: 0.9827427650769354
              precision    recall  f1-score   support

    negative       0.95      0.98      0.97     67963
     neutral       1.00      0.99      0.99    125212
    positive       0.99      0.98      0.98    124446

    accuracy                           0.98    317621
   macro avg       0.98      0.98      0.98    317621
weighted avg       0.98      0.98      0.98    317621
```



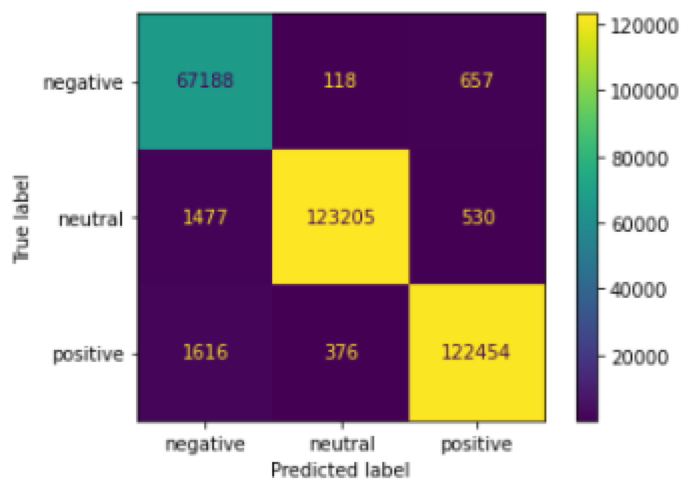


<u>With LDA</u>

```
the weighted f1_score: 0.98503143515179
              precision    recall  f1-score   support

    negative       0.96      0.99      0.97     67963
     neutral       1.00      0.98      0.99    125212
    positive       0.99      0.98      0.99    124446

    accuracy                           0.98    317621
   macro avg       0.98      0.99      0.98    317621
weighted avg       0.99      0.98      0.99    317621
```



Weighted f1-score table:

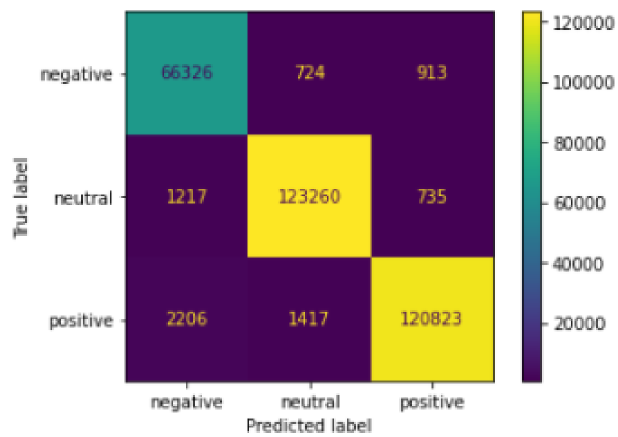
	TF-IDF	Bi-gram	Tri-gram	LDA
SVM	**0.984834**	**0.984421**	**0.982743**	**0.98503**

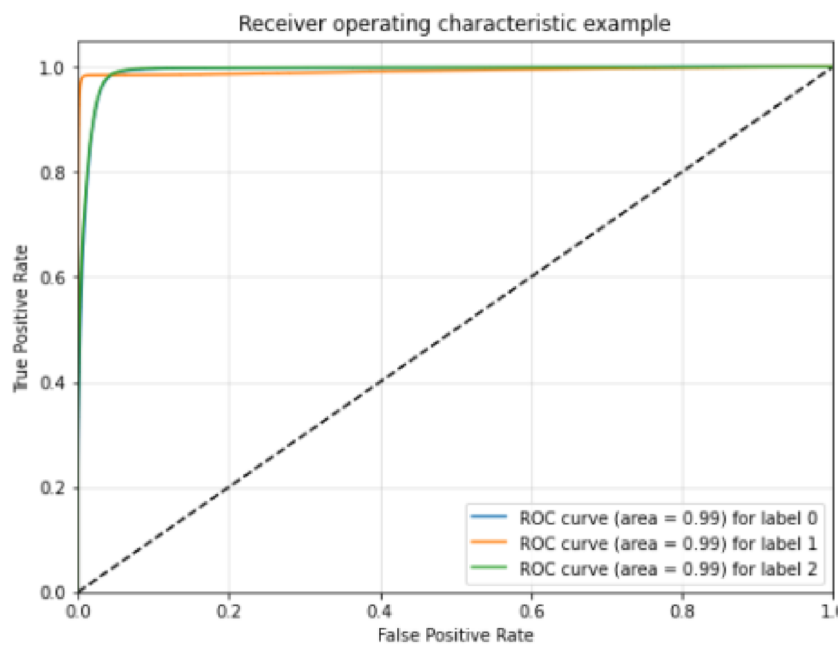
- <u>LR</u>

<u>With TF-IDF</u>

the weighted f1_score: 0.9773354871565509

	precision	recall	f1-score	support
negative	0.95	0.98	0.96	67963
neutral	0.98	0.98	0.98	125212
positive	0.99	0.97	0.98	124446
accuracy			0.98	317621
macro avg	0.97	0.98	0.98	317621
weighted avg	0.98	0.98	0.98	317621



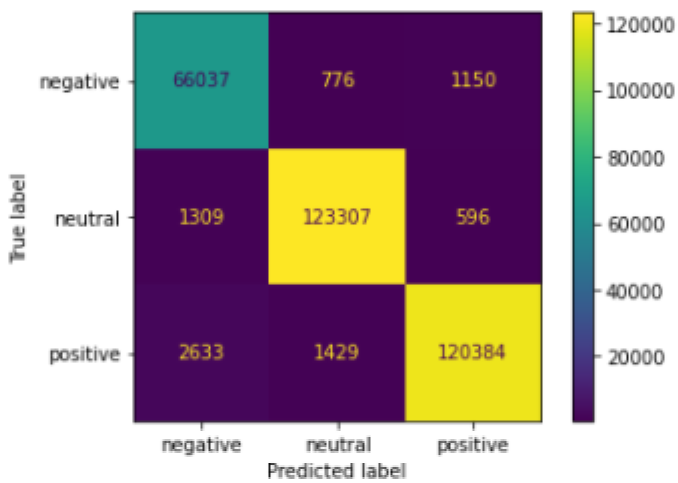


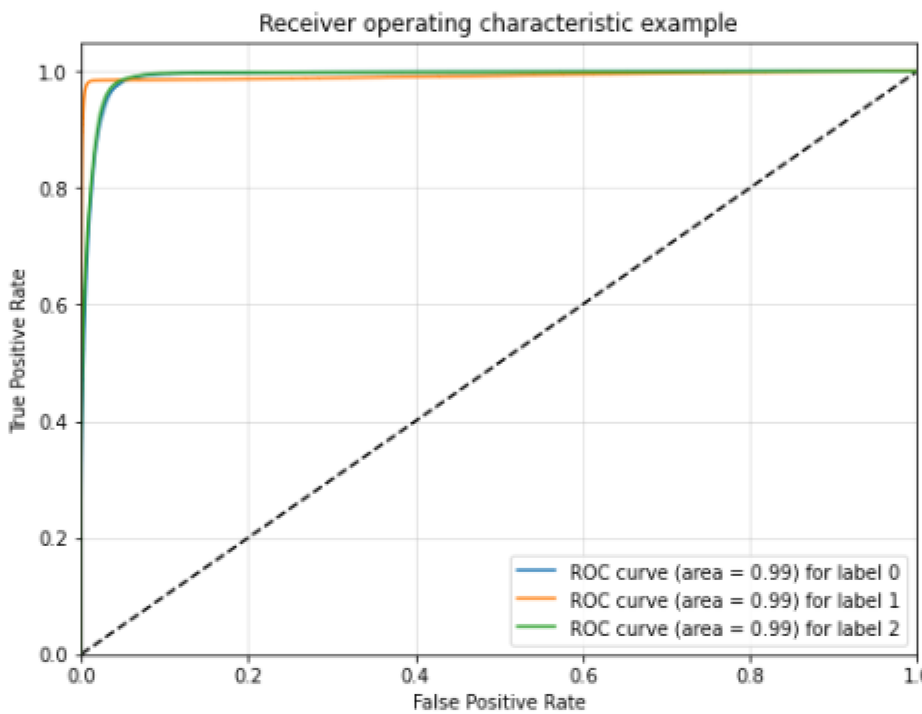
With bi-gram

```
the weighted f1_score: 0.9752064963678975
              precision    recall  f1-score   support

    negative       0.94      0.97      0.96     67963
     neutral       0.98      0.98      0.98    125212
    positive       0.99      0.97      0.98    124446

    accuracy                           0.98    317621
   macro avg       0.97      0.97      0.97    317621
weighted avg       0.98      0.98      0.98    317621
```

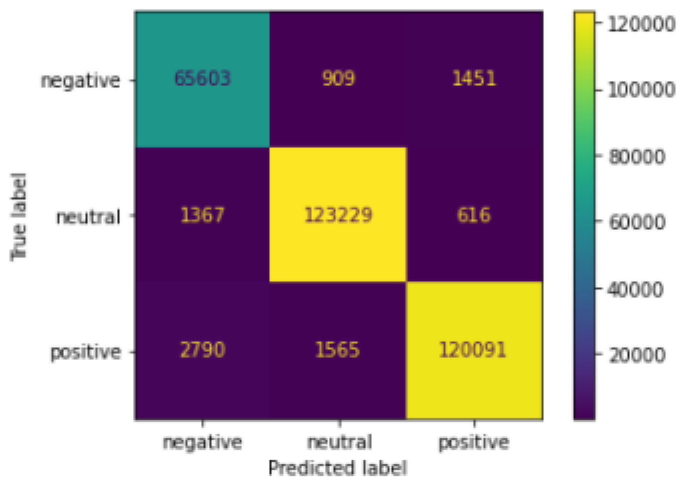


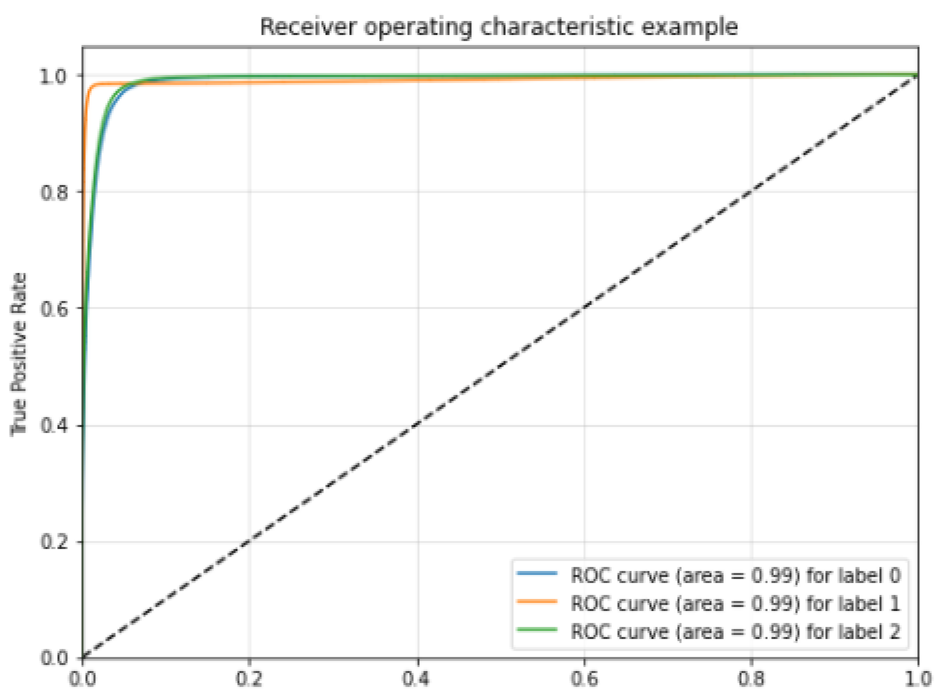


With tri-gram

the weighted f1_score: 0.972668953653931

	precision	recall	f1-score	support
negative	0.94	0.97	0.95	67963
neutral	0.98	0.98	0.98	125212
positive	0.98	0.97	0.97	124446
accuracy			0.97	317621
macro avg	0.97	0.97	0.97	317621
weighted avg	0.97	0.97	0.97	317621



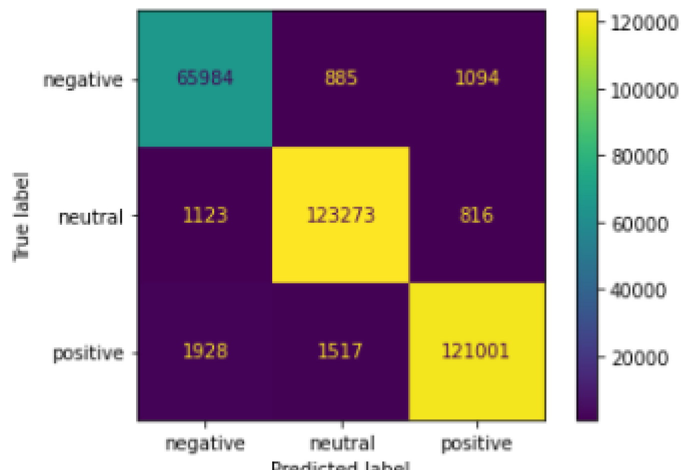


With LDA

```
the weighted f1_score: 0.9768405418674578
              precision    recall  f1-score   support

    negative       0.96      0.97      0.96     67963
     neutral       0.98      0.98      0.98    125212
    positive       0.98      0.97      0.98    124446

    accuracy                           0.98    317621
   macro avg       0.97      0.98      0.97    317621
weighted avg       0.98      0.98      0.98    317621
```



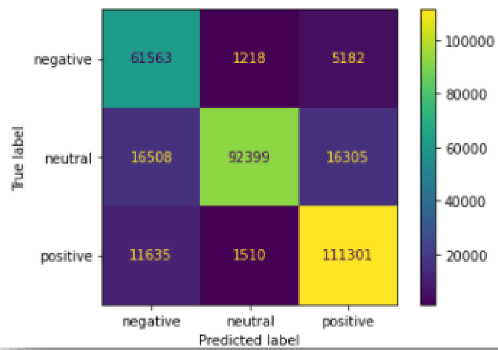
Weighted f1-score table:

	TF-IDF	Bi-gram	Tri-gram	LDA
LR	**0.977335**	**0.975206**	**0.972669**	**0.97684**

- **NB**
  With TF-IDF

the weighted f1_score: 0.8367833448616921

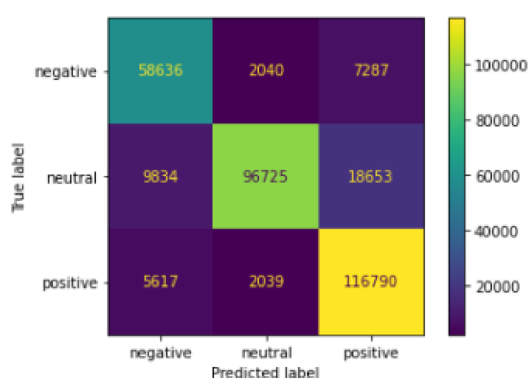
	precision	recall	f1-score	support
negative	0.69	0.91	0.78	67963
neutral	0.97	0.74	0.84	125212
positive	0.84	0.89	0.87	124446
accuracy			0.84	317621
macro avg	0.83	0.85	0.83	317621
weighted avg	0.86	0.84	0.84	317621



With Bi-gram

the weighted f1_score: 0.8566065501750139

	precision	recall	f1-score	support
negative	0.79	0.86	0.83	67963
neutral	0.96	0.77	0.86	125212
positive	0.82	0.94	0.87	124446
accuracy			0.86	317621
macro avg	0.86	0.86	0.85	317621
weighted avg	0.87	0.86	0.86	317621

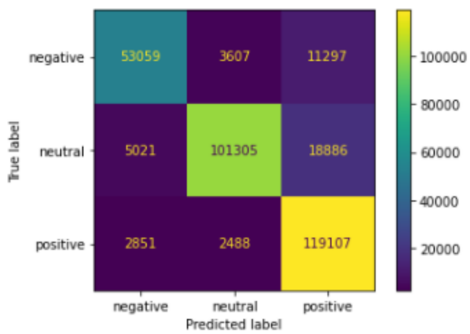


With Tri-gram

```
the weighted f1_score: 0.8605003974634658
              precision    recall  f1-score   support

    negative       0.87      0.78      0.82     67963
     neutral       0.94      0.81      0.87    125212
    positive       0.80      0.96      0.87    124446

    accuracy                           0.86    317621
   macro avg       0.87      0.85      0.85    317621
weighted avg       0.87      0.86      0.86    317621
```

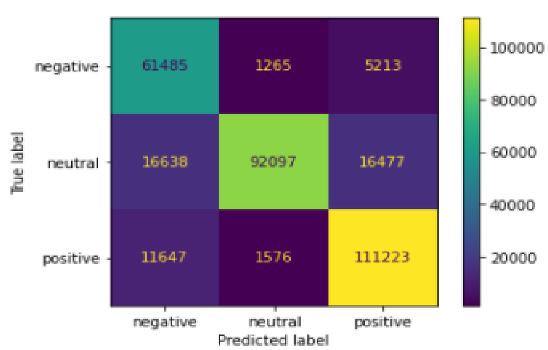


With LDA

```
the weighted f1_score: 0.8353040213382896
              precision    recall  f1-score   support

    negative       0.68      0.90      0.78     67963
     neutral       0.97      0.74      0.84    125212
    positive       0.84      0.89      0.86    124446

    accuracy                           0.83    317621
   macro avg       0.83      0.84      0.83    317621
weighted avg       0.86      0.83      0.84    317621
```



The Best models:

According to the weighted f1-score :

	TF-IDF	Bi-gram	Tri-gram	LDA
SVM	**0.984834**	**0.984421**	**0.982743**	**0.98503**
LR	**0.977335**	**0.975206**	**0.972669**	**0.97684**
NB	**0.836783**	**0.856607**	**0.860500**	**0.835304**

1. **svm with tf-idf**
2. **lr with tf-idf**
3. **Nb with tri-gram**

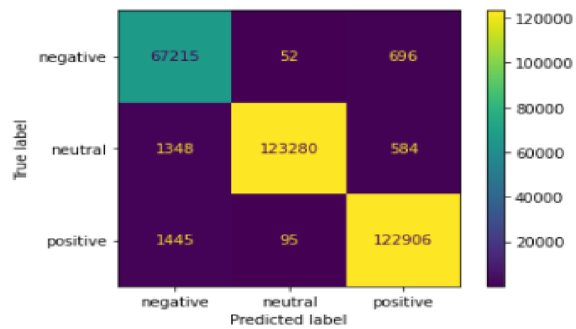
- **<u>Stacking</u>**

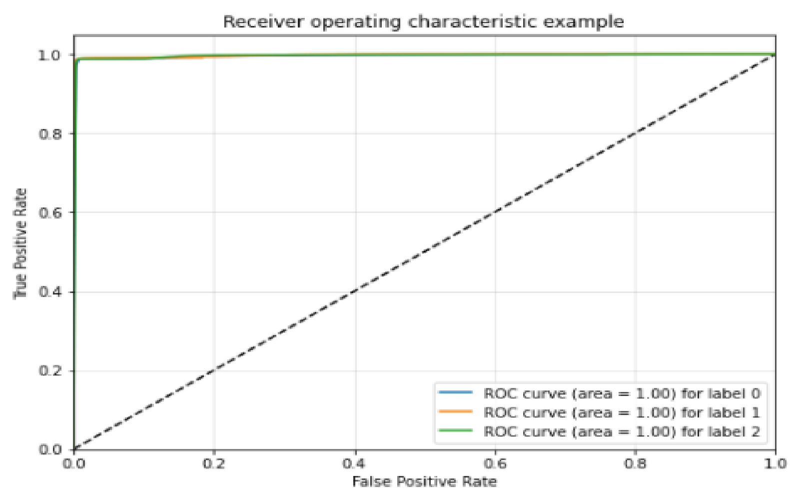
<u>(Aggregator svm)</u>

```
the weighted f1_score: 0.9867682084841153
              precision    recall  f1-score   support

    negative       0.96      0.99      0.97     67963
     neutral       1.00      0.98      0.99    125212
    positive       0.99      0.99      0.99    124446

    accuracy                           0.99    317621
   macro avg       0.98      0.99      0.98    317621
weighted avg       0.99      0.99      0.99    317621
```





The best ensemble model is Stacking Aggregator SVM Using the TF-IDF

Stacking Aggregator SVM	**0.9867**

Scraping data

Our MVP build on 5 topics (Hyundai, Peugeot, BMW, Mercedes, Kia)

This is our data frame

tweet_id	clean_text	author_id	clean_description
1446988499039801345	bmw italia spider	1294413026205073409	le monde change et nous devons changer avec lui
1446988474033352710	bmw	1275470896128458753	sourire aux levres pourtant interieur balafre
1446988313030578179	know pushing fin fin bmw got choke	918239639718199299	snapchat jgoble tiktok g blelee fb goble lee
1446988229530501121	bmw italia spider	573042416	nwarland
1446988082750832640	obtaining vehicles business name big flex usin...	1444427085066514433	serial entrepreneur business coach beauty infl...

I Transmitted it using tf-idf model

```
the shape of scraping tweets :  (106703, 276218)
```

using the champion model, I labelled the data.

```
neutral     56470
positive    37130
negative    13103
Name: sentiment, dtype: int64
```

This is the final data frame. Now, it is ready to visualization

tweet_id	clean_text	author_id	clean_description	sentiment
1446988499039801345	bmw italia spider	1294413026205073409	le monde change et nous devons changer avec lui	neutral
1446988474033352710	bmw	1275470896128458753	sourire aux levres pourtant interieur balafre	neutral
1446988313030578179	know pushing fin fin bmw got choke	918239639718199299	snapchat jgoble tiktok g blelee fb goble lee	neutral
1446988229530501121	bmw italia spider	573042416	nwarland	neutral
1446988082750832640	obtaining vehicles business name big flex usin...	1444427085066514433	serial entrepreneur business coach beauty infl...	neutral