



ELG5901

## **Beta Emotion Analysis**

**Uottawa supporter:**  
**Prof. Dr. Murat Simsek**

**Egypt Mentor:**  
**Dr. Mayada Hadhoud**

**Microsoft supporter:**  
**Eng. Michel Naim**

**Student Name:**  
Dina Abdelhady

## **Table of contents:**

<i>Abstract</i> -----	3
<i>Methodology</i> -----	3
<i>Dataset</i> -----	3
<i>Implementation</i> -----	4
<i>Result &amp; Analysis</i> -----	4

## **Abstract:**

This report aims to produce emotion analysis classification predictions and compare them and generate and communicate the insights and discuss the implementation steps of applying (building the model, transformation and evaluation, etc.) the strategy of the BETA was built on three of multiclass classification algorithms (random forest , multi-layer perceptron, xgboost) then building Ensemble model. I will introduce the detailed steps of implementation of this strategy and discuss the achieved results.

## **Methodology:**

I used (Emotion data set from Kaggle) for implementation. After the preprocessing step, I apply feature selection technique. Then, I transformed it to TF-IDF, Bi-gram and Tri-gram. Then I trained and tested the RF, MLP and XGB 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 Emotion dataset. It contains 20,000 tweets. The tweets have been annotated (Joy, Love, Surprise, Sadness, anger and fear). and they can be used to build the machine learning model able to detect Emotion.

## **Implementation:**

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

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

## **Result & Analysis:**

### **Emotion dataset dataset:**

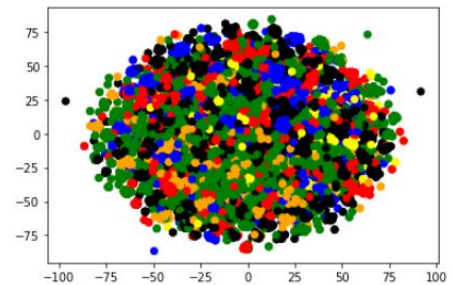
Let's take a look on Emotion dataset.

	text	Emotion
0	im feeling rather rotten so im not very ambiti...	sadness
1	im updating my blog because i feel shitty	sadness
2	i never make her separate from me because i do...	sadness
3	i left with my bouquet of red and yellow tulip...	joy
4	i was feeling a little vain when i did this one	sadness
5	i cant walk into a shop anywhere where i do no...	fear
6	i felt anger when at the end of a telephone call	anger
7	i explain why i clung to a relationship with a...	joy
8	i like to have the same breathless feeling as ...	joy
9	i jest i feel grumpy tired and pre menstrual w...	anger

```

joy          6761
sadness      5797
anger        2709
fear         2373
love         1641
surprise      719
Name: Emotion, dtype: int64

```



## Cleaning :

	text	Emotion	clean_text
0	im feeling rather rotten so im not very ambiti...	sadness	feeling rather rotten ambitious right
1	im updating my blog because i feel shitty	sadness	updating blog feel shitty
2	i never make her separate from me because i do...	sadness	never make separate ever want feel like ashamed
3	i left with my bouquet of red and yellow tulip...	joy	left bouquet red yellow tulips arm feeling sli...
4	i was feeling a little vain when i did this one	sadness	feeling little vain one
5	i cant walk into a shop anywhere where i do no...	fear	cant walk shop anywhere feel uncomfortable
6	i felt anger when at the end of a telephone call	anger	felt anger end telephone call
7	i explain why i clung to a relationship with a...	joy	explain clung relationship boy many ways immat...
8	i like to have the same breathless feeling as ...	joy	like breathless feeling reader eager see happe...
9	i jest i feel grumpy tired and pre menstrual w...	anger	jest feel grumpy tired pre menstrual probably ...

As a first step, I work on the imbalanced data. Then, I split the data to 80% train and 20% test.

```
y_train.value_counts()
```

```
joy          5390
sadness      4665
anger        2159
fear         1883
love         1314
surprise     589
```

```
y_test.value_counts()
```

```
joy          1371
sadness      1132
anger        550
fear         490
love         327
surprise     130
```

## **Feature engineering:**

### **TF-IDF**

```
X_train: (16000, 14674) X_test: (4000, 14674)
```

### **N-gram**

- **bi-gram**

```
X_train: (16000, 95781) X_test: (4000, 95781)
```

- **tri-gram**

```
X_train: (16000, 170371) X_test: (4000, 170371)
```

## **Build model:**

	RF_TF_IDF	MLP_TF_IDF	xgb_TF_IDF
<b>All features</b>	90.37	90.7	90.9
<b>10000</b>	90.37	90.34	<b>91.03</b>
<b>5000</b>	89.91	90.44	90.83
<b>1000</b>	89.21	90.13	90.28

	RF_ Bigram	MLP_ Bigram	xgb_ Bigram
<b>10000</b>	89.38	<b>90.94</b>	90.92
<b>5000</b>	88.4	90.39	90.3
<b>1000</b>	87.7	89.38	89.3

	RF_ Trigram	MLP_ Trigram	xgb_ Trigram
<b>20000</b>	89.7	90.4	<b>90.69</b>
<b>10000</b>	88.76	90.39	90.63
<b>5000</b>	89.04	90.3	90.09
<b>1000</b>	89.34	88.1	89.35

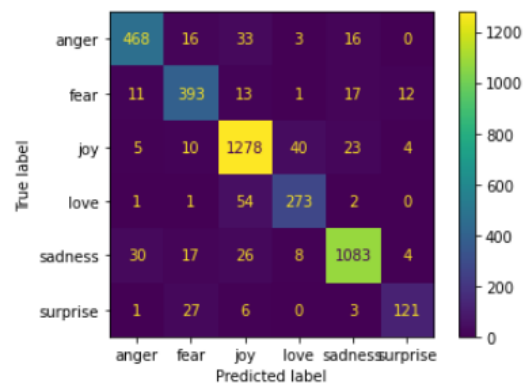
- **Best RF**

```
the weighted f1_score: 0.9037784528820645
precision    recall  f1-score   support

   anger      0.91      0.87      0.89       536
    fear      0.85      0.88      0.86       447
     joy      0.91      0.94      0.92      1360
    love      0.84      0.82      0.83       331
  sadness      0.95      0.93      0.94      1168
   surprise      0.86      0.77      0.81       158

 accuracy          0.90      4000
macro avg      0.88      0.87      0.88      4000
weighted avg      0.90      0.90      0.90      4000
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/depreca
warnings.warn(msg, category=FutureWarning)
```



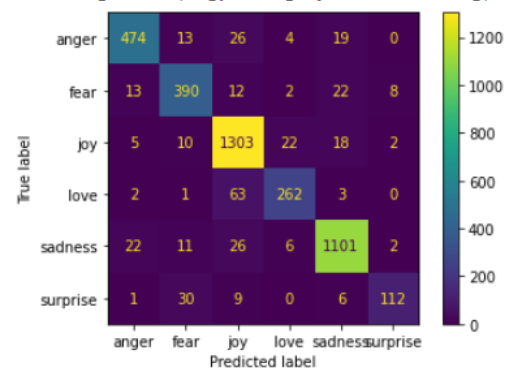
- **Best MLP**

```
the weighted f1_score: 0.9094937516143247
precision    recall  f1-score   support

   anger      0.92      0.88      0.90       536
    fear      0.86      0.87      0.86       447
     joy      0.91      0.96      0.93      1360
    love      0.89      0.79      0.84       331
  sadness      0.94      0.94      0.94      1168
   surprise      0.90      0.71      0.79       158

 accuracy          0.91      4000
macro avg      0.90      0.86      0.88      4000
weighted avg      0.91      0.91      0.91      4000
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/d
warnings.warn(msg, category=FutureWarning)
```

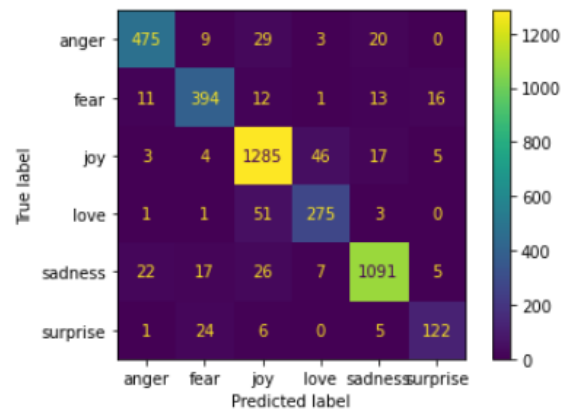


- **Best Xgb**

```
the weighted f1_score: 0.9103286474103371
```

	precision	recall	f1-score	support
anger	0.93	0.89	0.91	536
fear	0.88	0.88	0.88	447
joy	0.91	0.94	0.93	1360
love	0.83	0.83	0.83	331
sadness	0.95	0.93	0.94	1168
surprise	0.82	0.77	0.80	158
accuracy			0.91	4000
macro avg	0.89	0.87	0.88	4000
weighted avg	0.91	0.91	0.91	4000

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/d
warnings.warn(msg, category=FutureWarning)
```





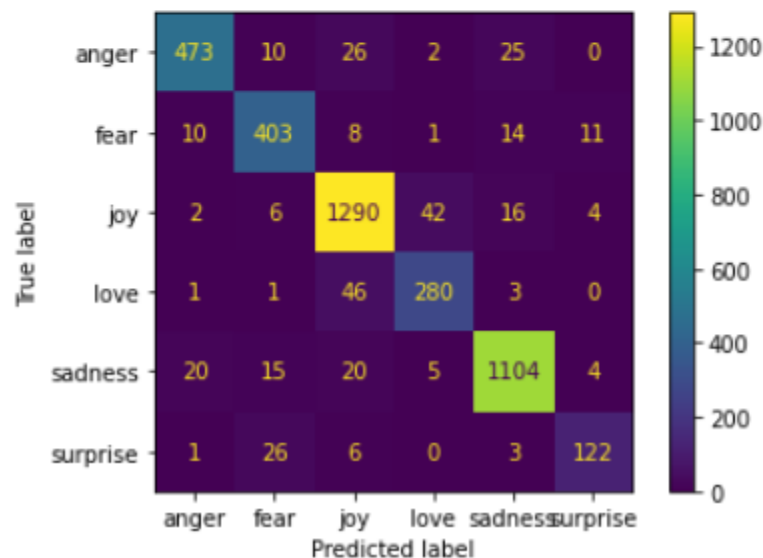
- **Stacking**

**Voting: 91.77%**

the weighted f1\_score: 0.9177167521395247

	precision	recall	f1-score	support
anger	0.93	0.88	0.91	536
fear	0.87	0.90	0.89	447
joy	0.92	0.95	0.94	1360
love	0.85	0.85	0.85	331
sadness	0.95	0.95	0.95	1168
surprise	0.87	0.77	0.82	158
accuracy			0.92	4000
macro avg	0.90	0.88	0.89	4000
weighted avg	0.92	0.92	0.92	4000

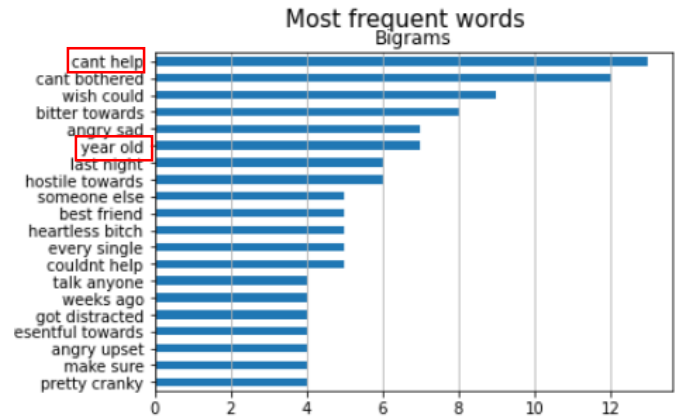
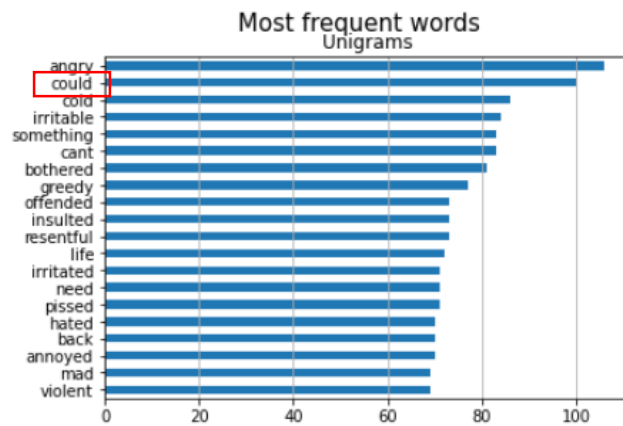
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecate\_warnings.warn(msg, category=FutureWarning)



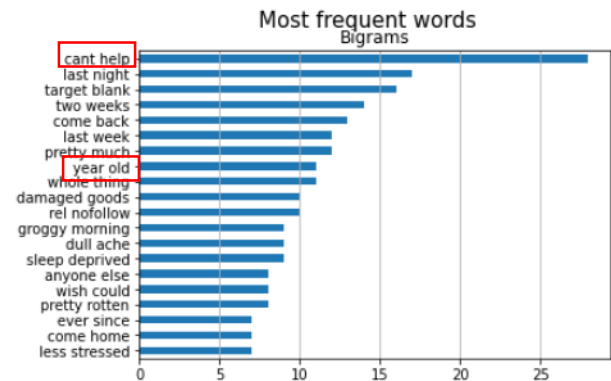
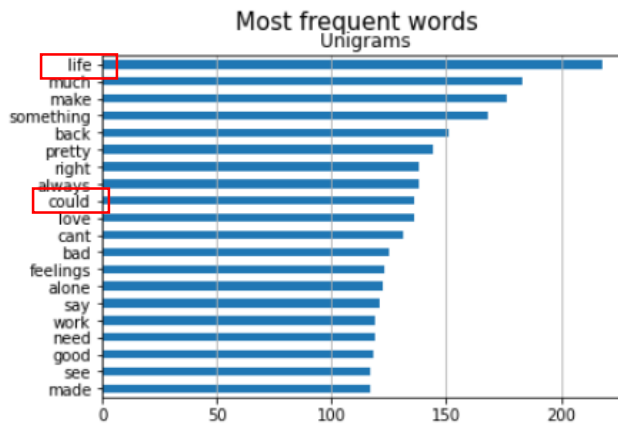
## **Error analysis**

Still There are some common words between the three class that make the model confused.

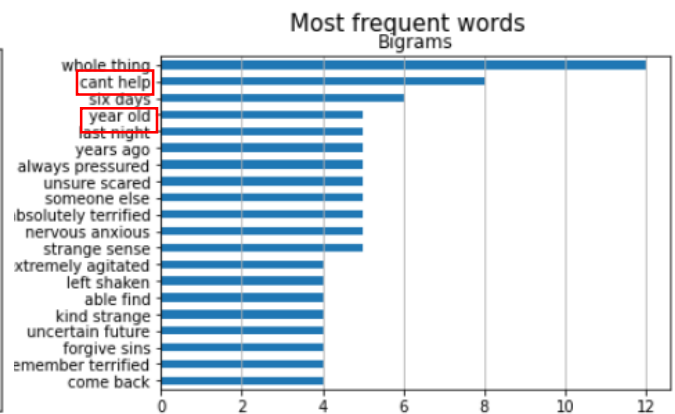
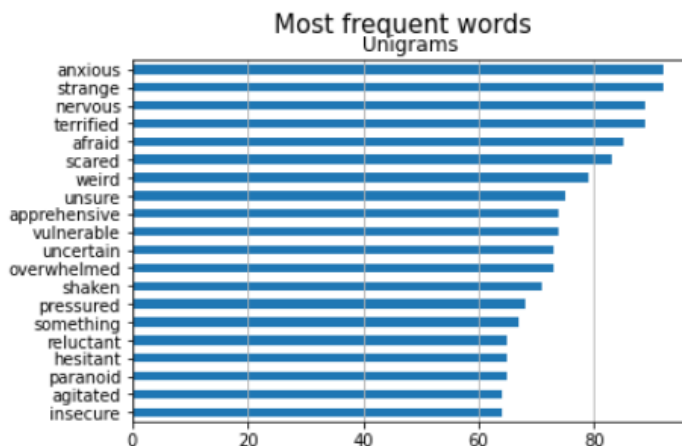
## Anger:



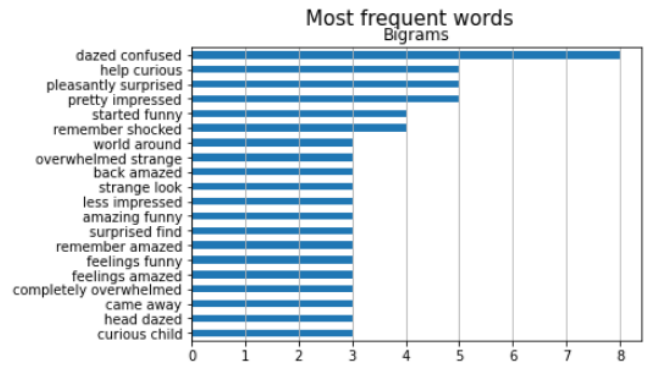
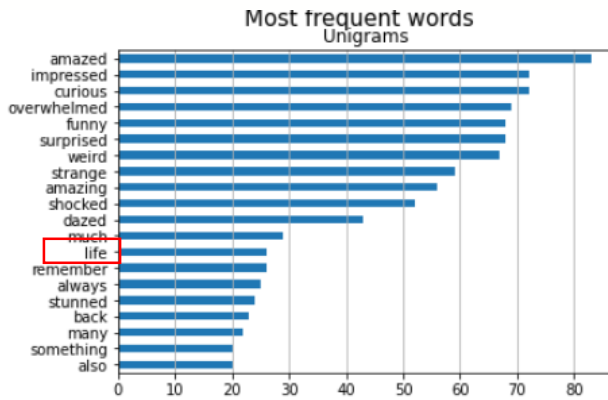
## Sadness:



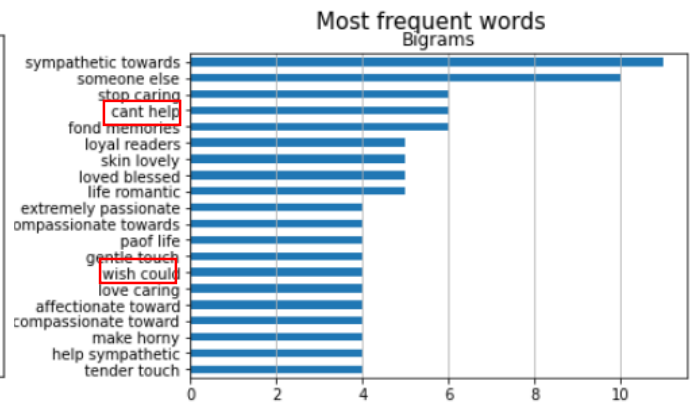
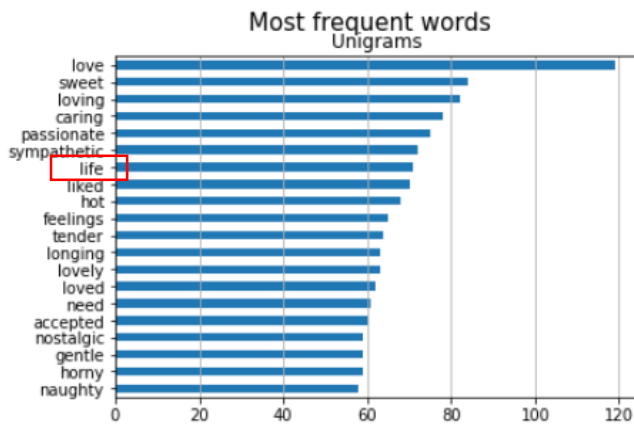
## Fear:



## Surprise:



## Love:



Apply oversample technique to balance the training data.

```

anger      5410
fear       5410
sadness    5410
joy        5410
love       5410
surprise   5410
Name: emotion, dtype: int64

```

Voting ensembl model :

```
the weighted f1_score: 0.9151108536062084
```

	precision	recall	f1-score	support
anger	0.92	0.89	0.90	536
fear	0.87	0.88	0.88	447
joy	0.94	0.93	0.94	1360
love	0.79	0.92	0.85	331
sadness	0.95	0.94	0.94	1168
surprise	0.80	0.83	0.81	158
accuracy			0.91	4000
macro avg	0.88	0.90	0.89	4000
weighted avg	0.92	0.91	0.92	4000

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/d  
warnings.warn(msg, category=FutureWarning)
```

