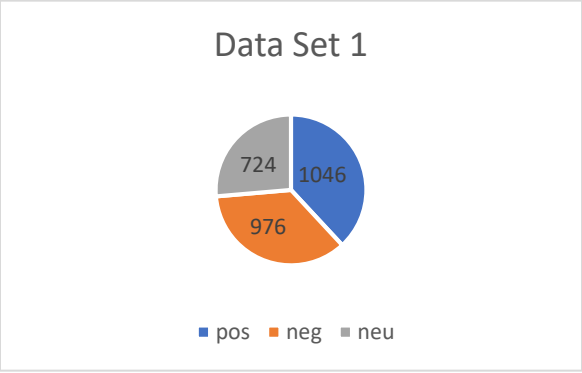
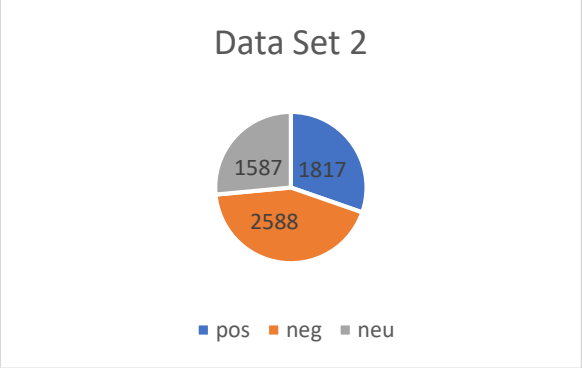
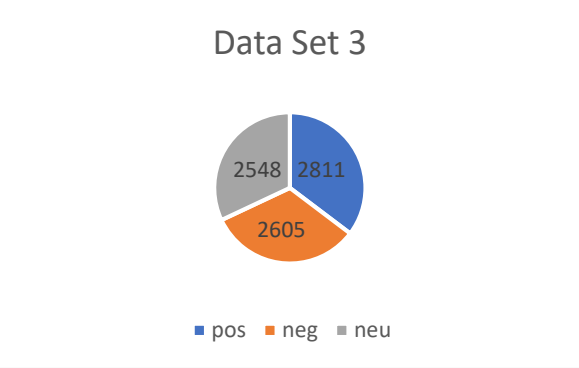
# Introduction

According to the large number of Arabic users in the social media those days and after the Revolutions in the Arab World it is found that it is important to build sentiment analysis models for Arabic tweets to analyze Twitter comments as having positive, negative or neutral sentiments. We have tried three different machine learning algorithms to build models for sentiment analysis. The three different algorithms are known as: **Gaussian Naive Bayes, multinomial Naive Bayes and Support vector machines (SVMs).**

The training part is done on three different data sets. Each one with larger number of tweets as shown in the following figures.



We Used scikit-learn which is simple and efficient tools for data mining and data analysis. For training the tweets we choose two different ways to change text to new shape that can be trained using the discussed classifiers. **The two different ways are CountVectorizer and TfidfVectorizer.**

- CountVectorizer: Convert a collection of text documents to a matrix of token counts

- TfidfVectorizer: Convert a collection of raw documents to a matrix of TF-IDF features. Equivalent to CountVectorizer followed by TfidfTransformer.

The important part before training or testing is to preprocess the tweets to clean it and remove unnecessary words from it. So, we will discuss now the preprocessing techniques that we used.

**Preprocessing Steps**

1. remove links (any sword that starts with ‘http’ or ‘www’)
2. remove # character #messi'-->'messi'
3. remove mentions (@messi)
4. remove consecutive characters from a string ('شكرا','شكراااااا')
5. remove Arabic stop words
6. stem words using Dr. Samhaa light stemmer
7. remove special characters
8. remove words length less than 2 characters
9. remove words that its start or end character is a number

Now we will show the accuracy of each dataset using the preprocessing with the two ways CountVectorizer and TfidfVectorizer.

Test 1: First we start with those preprocessing:

1. remove links (any sword that starts with ‘http’ or ‘www’)
2. remove # character #messi'-->'messi'
3. remove mentions (@messi)
4. remove consecutive characters from a string ('شكرا','شكراااااا')

## Data Set 1

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.5007 | 0.559 | 0.575 |
| CountVectorizer | 0.489 | 0.588 | 0.544 |

## Data Set 2

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.477 | 0.532 | 0.576 |
| CountVectorizer | 0.489 | 0.588 | 0.544 |

## Data Set 3

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.475 | 0.556 | 0.546 |
| CountVectorizer | 0.483 | 0.562 | 0.527 |

Test 2: ADD Arabic Stop Words

## Data Set 1

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.5007 | 0.565 | 0.571 |
| CountVectorizer | 0.489 | 0.590 | 0.538 |

## Data Set 2

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.478 | 0.534 | 0.568 |
| CountVectorizer | 0.489 | 0.590 | 0.538 |

## Data Set 3

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.475 | 0.557 | 0.554 |
| CountVectorizer | 0.483 | 0.557 | 0.525 |

Test 3: remove special characters, remove words length less than 2 characters and remove words that its start or end character is a number.

## Data Set 1

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.474 | 0.544 | 0.537 |
| CountVectorizer | 0.445 | 0.550 | 0.496 |

## Data Set 2

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.468 | 0.543 | 0.527 |
| CountVectorizer | 0.445 | 0.550 | 0.496 |

## Data Set 3

| Method | Classifier | | |
|---|---|---|---|
| | Gaussian Naïve Bayes | Multinomial Naïve Bayes | SVM |
| TfidfVectorizer | 0.436 | 0.531 | 0.513 |
| CountVectorizer | 0.440 | 0.551 | 0.489 |

Note: when we used SVM classifier without kernel the accuracy was very low around 37%, as to change non-linear separable data to linear separable ones with more dimensions there is a trick called kernel in SVM classifier.

We found someone who try an algorithm that depend on calculating probability of positive, negative and neutral words in data set and using this probability, he use some formulas to make classification decision based on which value is greater. We modify it to fit us and we calculate the accuracy of that classifier and compare the result with the three other classifiers. This new classifier shows accuracy equal to 51.2%

First when we start the project we used an algorithm that depend on extracting features manually and train it is using NLTK library. This classifier gives us 39.3% accuracy which was so bad and that's why we learn how to use sklearn and search for more than one technique.

After testing we found that **Multinomial Naïve Bayes** is the best classifier in case of classifying Arabic tweets compared to the other two classifiers.