

August 03, 2020

Data Science with Python

Jyoti Sharma

# Sentiment Analysis

# Data Classification Project

**Table of Contents**

[Data Classification Goal 2](#_Toc48163433)

[Process Flow 4](#_Toc48163434)

[Test Flow Diagram 4](#_Toc48163435)

[Classification Process 5](#_Toc48163436)

[Most used words in feature set 1 5](#_Toc48163438)

[Most used words in feature set 2 6](#_Toc48163439)

[Classifiers used 7](#_Toc48163440)

[How Voted Classifier works? 7](#_Toc48163441)

[Results 8](#_Toc48163442)

[Accuracy achieved with feature set 8](#_Toc48163443)

[Accuracy achieved with feature set 9](#_Toc48163445)

[Triggering Execution 10](#_Toc48163447)

[Additional Packages and Purpose 10](#_Toc48163448)

[Directory structure 10](#_Toc48163449)

[Steps to run 10](#_Toc48163450)

[Bibliography 12](#_Toc48163451)

# **Data Classification Goal**

The goal of this project is to find the best classifier to perform sentiment analysis of tweets. The goal is accomplished using a tool kit called nltk (Natural Language Toolkit). The tweets are classified into two categories ‘Positive’ or ‘Negative’.

Distribution of Data

Tweets are briefly distributed into two categories. The train dataset consists of ﻿﻿26911 positive and ﻿﻿23072 Negative tweets, the test dataset consists of ﻿﻿29546 positive and ﻿﻿﻿20460 Negative tweets

**Train tweets distribution:**

A screenshot of a cell phone

Description automatically generated

**Test tweets distribution:**

A screenshot of a cell phone

Description automatically generated

**Understanding Data:**

The average length of tweets is approximately ﻿72 characters, the distribution looks like below

A close up of a piece of paper

Description automatically generated

# **Process Flow**

# Test Flow Diagram

A screenshot of a cell phone

Description automatically generated

# **Classification Process**

# In classification process for text feature sets play critical role, so in order to fine tune or understand its impact I have carried out classification in two ways

1. Include adjectives, verbs and noun as allowed words for constructing feature set.
2. Allow only adjectives and verbs for constructing feature set.

## Most used words in feature set 1

(allowed words: adjectives, verbs and noun)

A screenshot of a cell phone

Description automatically generated

## Most used words in feature set 2

(allowed words: adjectives and verbs)

A screenshot of a cell phone

Description automatically generated

# **Classifiers used**

For performing the prediction/classification, I have used below inbuilt classifiers

1. ﻿**MultinomialNB** classifier from ﻿**SklearnClassifier.**
2. ﻿**LogisticRegression** classifier from ﻿**SklearnClassifier.**
3. ﻿**SGD** Classifier classifier from ﻿**SklearnClassifier.**
4. ﻿**AdaBoostClassifier** classifier using ﻿**LinearSVC** as base learner from **SklearnClassifier.**
5. **AdaBoostClassifier** classifier using ﻿ **LogisticRegression** as base learner from **SklearnClassifier**
6. **Voted Classifier** which combines all 5 classifiers and classifies on the basis of votes assigned by each classifier. Every classifier is given equal weightage, labels of testing dataset are predicted by Voted classifier only.

## How Voted Classifier works?

A close up of a device

Description automatically generated

Every classifier vote has equal weightage.

# **Results**

## Accuracy achieved with feature set

## (allowed words: adjectives, verbs and noun)

**Accuracies for classifiers**

**A screenshot of a cell phone

Description automatically generated**

## Accuracy achieved with feature set

## (allowed words: adjectives, verbs)

A screenshot of a cell phone

Description automatically generated

# **Triggering Execution**

## Additional Packages and Purpose

In order to execute this suite, we need below additional packages

1. **NLTK** – Natural Language Processing ToolKit
2. **Plotnine** – For enabling use of ggplot functions for plots

## Directory structure

**Datasets:** Folder containing train and test datasets

**Plots:** Used for storing graphs (Graphs used in report)

**Pickled\_files:** Folder for storingTrained classifier pickle files, generated at run time.

**output XML**: Stored the execution console for one of the executions

A close up of a keyboard

Description automatically generated

## Steps to run

**Step 1:** Train the models (**script:** Train\_learners\_V\_1.1.py)

The script is used for train the classifiers and save them in pickle format inside **Pickled\_files**.

In order to train classifiers with allowed words as **adjectives and verbs**, value of allowed words need to be set as below:



In order to train classifiers with allowed words as **adjectives, verbs and Noun**, value of allowed words need to be set as below:

﻿allowed\_word\_types = ["J", "V", "RB", "N"]

**Note:** You can skip the execution of train, as pickled trainers, document, feature set for config <allowed words **adjectives and verbs>** is already present in Pickled\_files folders.

**Step 2:** Load the data (**script:** load\_pickled\_mod.py)

This file loads Load the pickled classifiers, documents and feature sets created in step 1.

**Step 3:** Predict/Test (**script:** Main\_run.py)

This is the main driver script which is predicting the sentiments using the Voted classifier.

If we want to re generate the plots we can trigger “Visual Data Analysis.py” script after triggering “Main\_run.py”.

(Note:All the plots are generated and stored in Plots folder)

# Bibliography

* Data source: <https://www.kaggle.com/>
* Nltk : <http://www.nltk.org/>
* Plotnine : <https://plotnine.readthedocs.io/en/stable/generated/plotnine.ggplot.html>
* <https://scikit-learn.org/>