

# **PROJECT REPORT**

**ON**

## **Text Classification**

**UNDER**

**HPKVN Sponsored Training on  
AI with Python  
23/02/2023 to 24/05/2023**

**Submitted by  
Vivek Sharma  
Vishal Negi**

**Project Mentor Name**

**Mrs. Himani**



**Education & Training Division (ETD)  
Centre for Development of Advanced Computing (C-DAC)  
(Ministry of Electronics & Information Technology, Govt. of India)  
A-34, Phase-VIII, Industrial Area, Mohali (160071)**

## TABLE OF CONTENTS

	Page No
<b>1 Introduction</b>	<b>3</b>
<b>2 Problem statement</b>	<b>4</b>
<b>3 Objectives</b>	<b>5</b>
<b>4 Design Methodology (Block Diagram or Work flow)</b>	<b>6</b>
<b>5 Dataset Explanation</b>	<b>8</b>
<b>6 Results – Front end clips, back end design clips, data base etc</b>	<b>12</b>
<b>7 Conclusion</b>	<b>13</b>
<b>8 References</b>	<b>14</b>

## Introduction

Text classification, also known as text categorization, is a natural language processing (NLP) task that involves automatically assigning predefined categories or labels to text documents based on their content.

The goal of text classification is to analyze and organize large volumes of unstructured text data by grouping similar documents together.

## Text Classification

“The user interface is quite straightforward and easy to use.”

Classifier takes the text as an input, analyzes its content, and then automatically assigns relevant tags:



In this project, we explore the task of text classification for review classification. The objective of this project is to build a model that can accurately classify reviews as positive or negative based on their content.

## **Problem statement**

The objective of this project is to develop a text classification model for review classification. The problem at hand involves analyzing a large number of customer reviews of a product and accurately classifying them as positive or negative based on the sentiment expressed in the text.

The ability to automatically classify reviews can provide valuable insights to businesses, allowing them to monitor customer sentiment and make data-driven decisions to enhance their products and services.

The challenge lies in building a model that can effectively capture the underlying sentiment in the text and generalize well to new, unseen reviews. The project aims to address this challenge by exploring and implementing machine learning techniques for review classification and evaluating their performance on a labeled dataset of customer reviews.

## Objective

To describe the dataset used for the review classification task, including its size, source, and any preprocessing steps applied.

To explain the machine learning models used for review classification and how they were trained and evaluated.

To compare the performance of the different machine learning models in terms of accuracy, precision, recall, and F1-score.

To identify any challenges or limitations encountered in the review classification task and discuss potential solutions or areas for future research.

To provide insights into the sentiment expressed in the customer reviews and how it can be used to improve the product or service being reviewed.

To highlight the potential benefits of using machine learning models for review classification and their applications in real-world scenarios.

Overall, the objectives of a report on review classification are to present the findings of the project and provide a clear understanding of the methodology used, the results achieved, and the implications of the findings.

## Design Methodology (Block Diagram or Work flow)

Designing a methodology for movie review sentiment analysis involves several steps. Here's a suggested framework:

**Define the Objectives:** Clearly state the goals and objectives of your movie review sentiment analysis. Determine whether you want to classify reviews as positive, negative, or neutral or if you want to assign a sentiment score to each review.

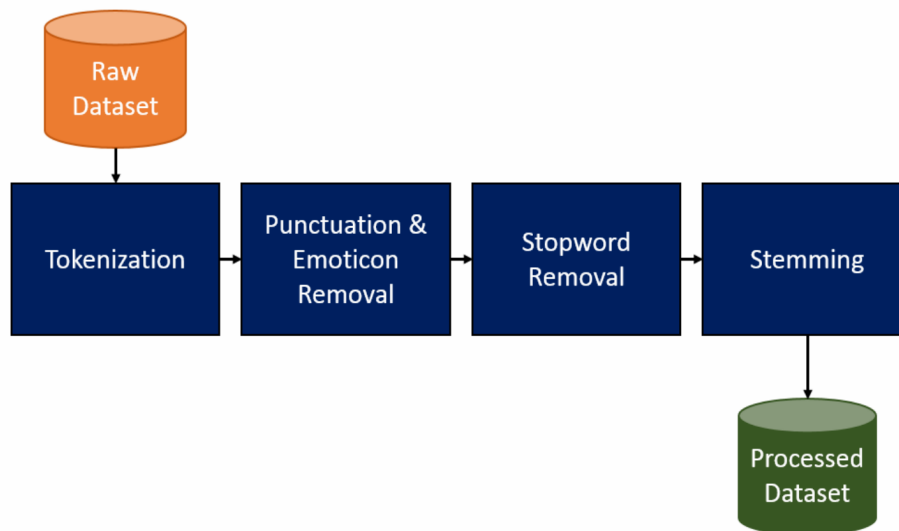
**Data Collection:** Gather a diverse and representative dataset of movie reviews. Include reviews from various sources such as online platforms, social media, blogs, or curated databases. Ensure the dataset covers different genres, time periods, and demographic perspectives.

**Preprocessing:** Clean and preprocess the collected data. Remove noise, formatting inconsistencies, and irrelevant information. Common preprocessing steps include lowercasing, removing punctuation, tokenization, and removing stop words. Consider stemming or lemmatization to normalize words.

**Labeling or Annotation:** Annotate the movie reviews with sentiment labels. This can be done manually by having human annotators assign positive, negative, or neutral labels to each review. Alternatively, you can use pre-labeled datasets or employ semi-supervised learning techniques to reduce the labeling effort.

**Feature Extraction:** Extract relevant features from the reviews to represent them numerically. Common approaches include using bag-of-words representation, TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings like Word2Vec or GloVe. These features will serve as input to your sentiment classification model.

**Model Selection and Training:** Choose a suitable machine learning model for sentiment classification. Common models include logistic regression, support vector machines (SVM), random forests, or more advanced models like recurrent neural networks (RNNs) or transformers such as BERT. Train the model on the labeled data, using a training-validation split or cross-validation.



**Model Evaluation:** Evaluate the performance of your sentiment classification model. Use evaluation metrics such as accuracy, precision, recall, F1 score, or area under the receiver operating characteristic curve (AUC-ROC) to measure its effectiveness. You can also conduct error analysis to understand the types of misclassifications the model makes.

**Fine-tuning and Optimization:** Refine and optimize your sentiment classification model based on the evaluation results. Experiment with different hyperparameters, model architectures, or feature representations to improve the performance. Consider techniques like cross-validation, grid search, or Bayesian optimization for parameter tuning.

**Deployment and Application:** Deploy your trained model to predict sentiment on new, unseen movie reviews. Implement an interface or API to accept user inputs and provide sentiment predictions. Monitor the model's performance over time and update it periodically to maintain accuracy.

**Limitations and Future Work:** Discuss the limitations of your sentiment analysis methodology, such as biases in the dataset or challenges in handling sarcasm or irony. Suggest areas for future improvement, such as incorporating contextual information or exploring ensemble methods to boost performance.

## Dataset Explanation

The dataset used for model building is “IMDB 50k Reviews”. The dataset consists of two columns and 50,000 rows.

**Review :** It is the column in which the thoughts of a viewer of movies are contained.

**Sentiment :** It is the column in which the reviews made by reviewers are classified manually as positive or negative for the movies.

```
In [8]: def clean(text):
        text=text.lower()
        text=re.sub('\[.*?\]', '', text)
        text=re.sub('<.*?>', '', text)
        text=re.sub('https?://\S+|www\.\S+', '', text)
        text=re.sub('%s' % re.escape(string.punctuation), '', text)
        text=re.sub('\w*\d\w*', '', text)
        text=re.sub('\n', '', text)
        return text
df['clean_text'] = df['review'].apply(lambda x: clean(x))
df.head()
```

Out[8]:

	review	sentiment	clean_text
0	One of the other reviewers has mentioned that ...	positive	one of the other reviewers has mentioned that ...
1	A wonderful little production.   The...	positive	a wonderful little production the filming tech...
2	I thought this was a wonderful way to spend ti...	positive	i thought this was a wonderful way to spend ti...
3	Basically there's a family where a little boy ...	negative	basically theres a family where a little boy j...
4	Petter Mattei's "Love in the Time of Money" is...	positive	petter matteis love in the time of money is a ...

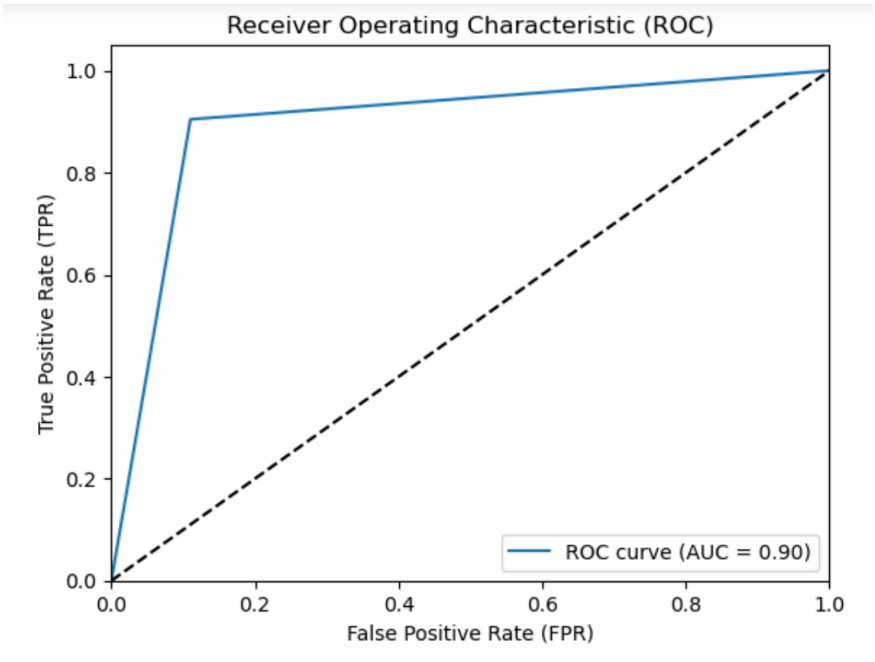
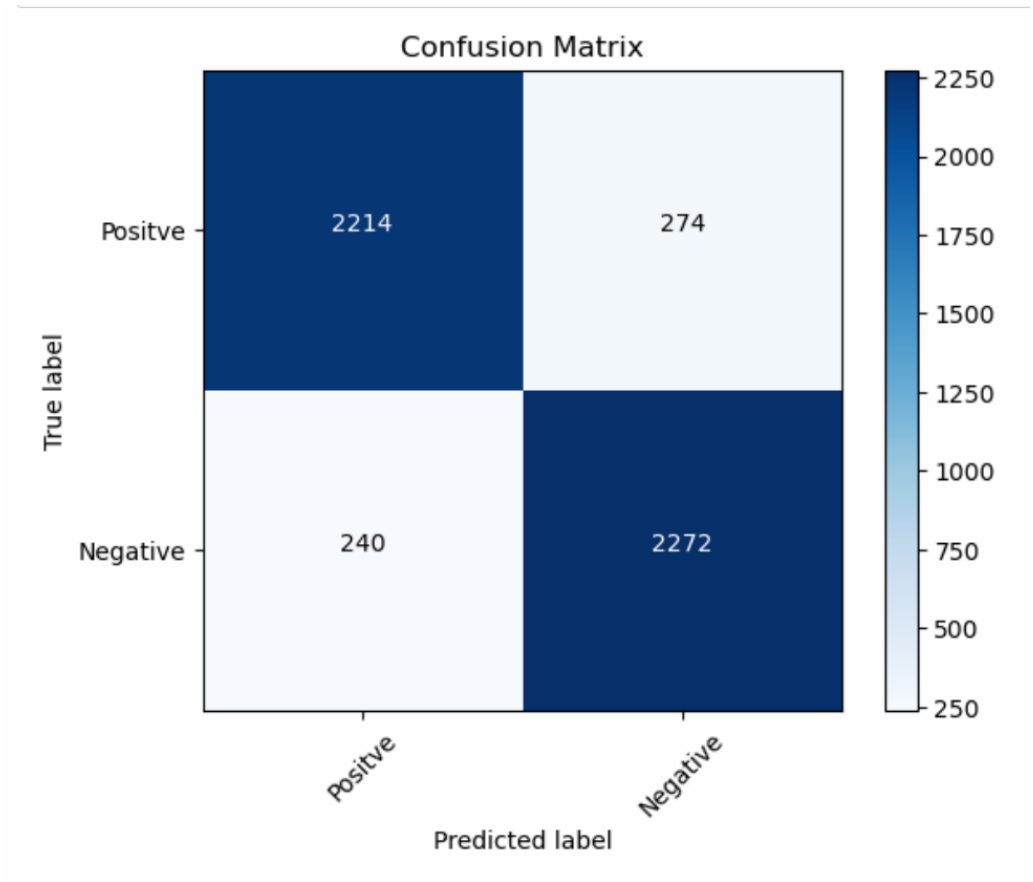
It contains 50,000 movie reviews (25,000 in training and 25,000 in testing) from IMDB, as well as each movie review’s binary sentiment: positive or negative.

The raw data contains the text of each movie review, and it has to be pre-processed before being fitted with any machine learning models.

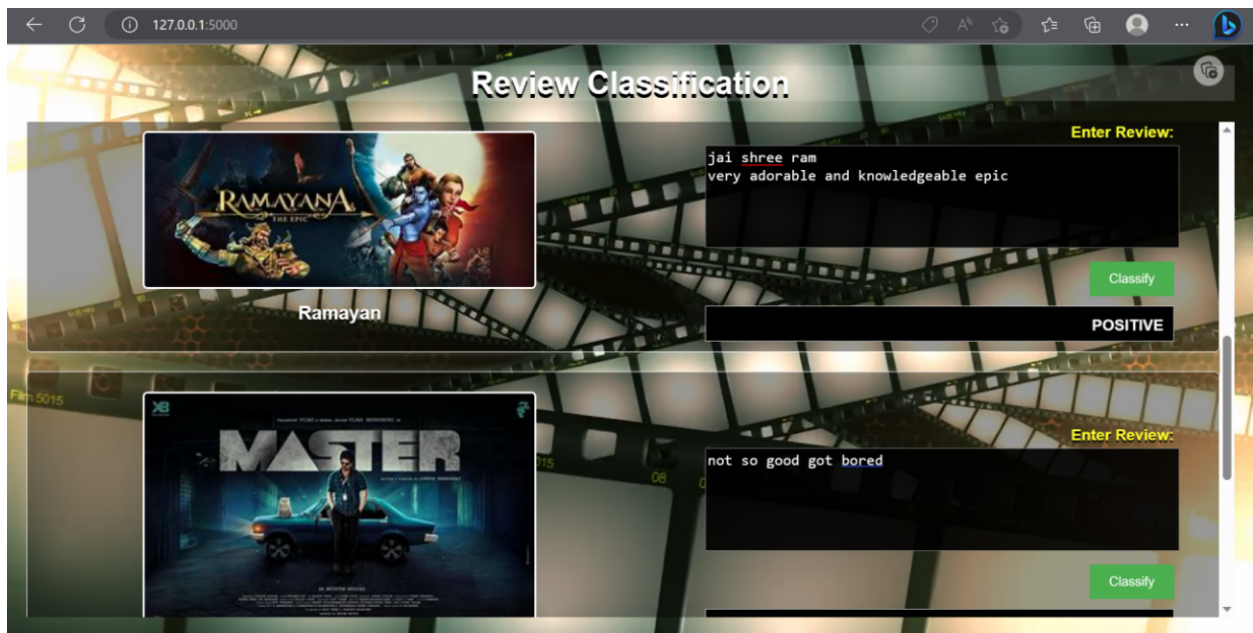
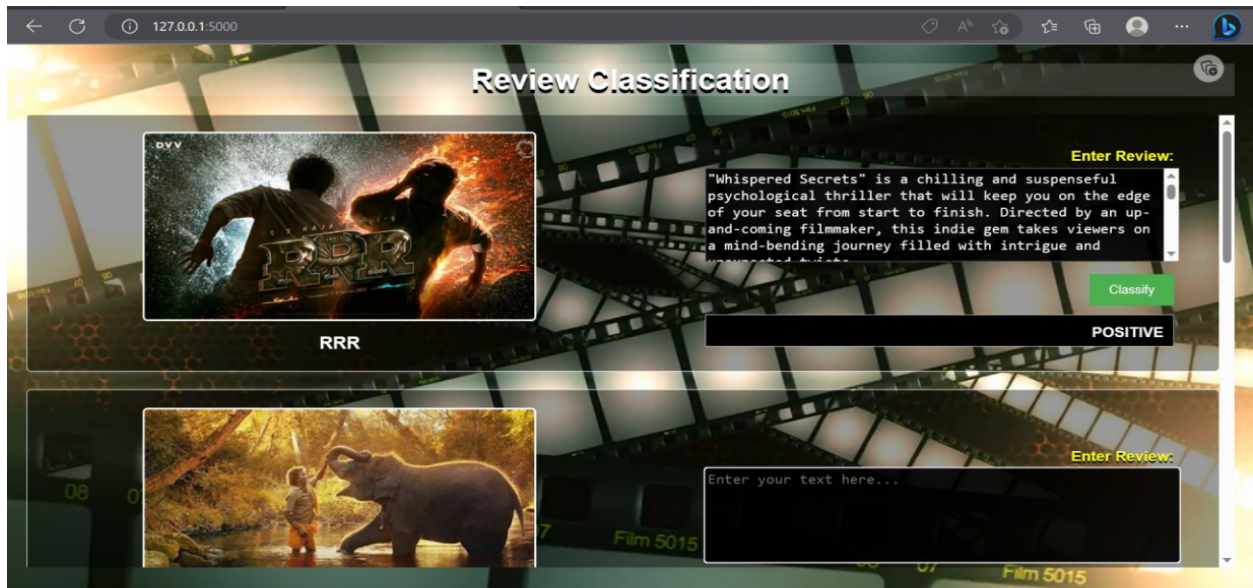
By using Keras’s built-in functions, we can easily get the processed dataset (i.e., a numerical data frame) for machine learning algorithms.



# Results



## Front end clips:



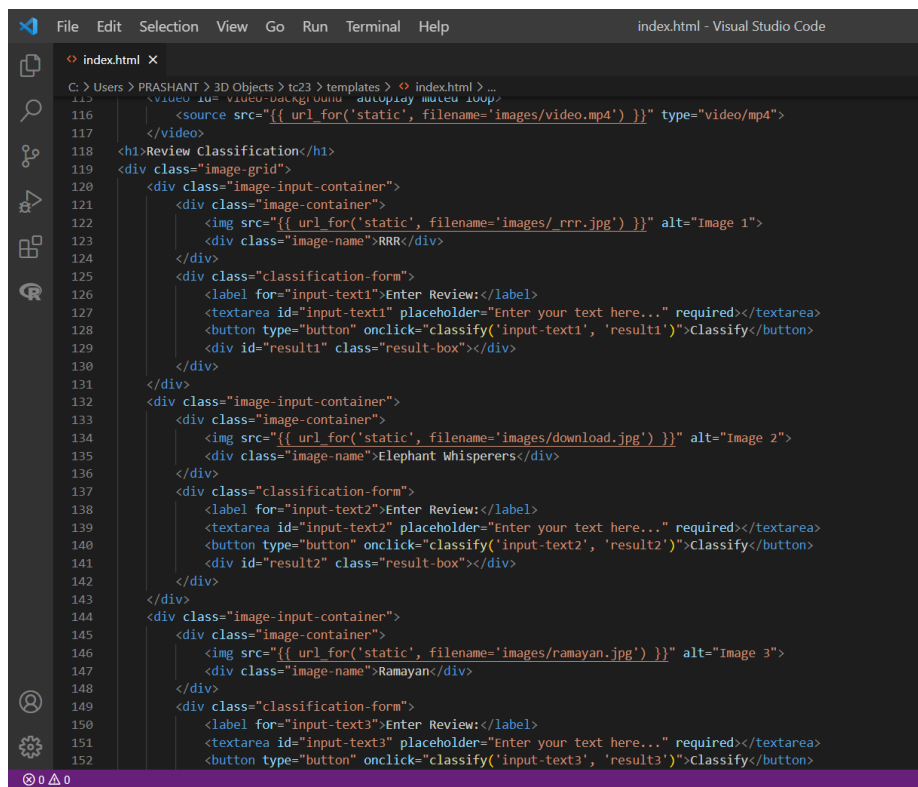
## Back end design clips:

```
Command Prompt - python app.py
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PRASHANT>cd "3D Objects"

C:\Users\PRASHANT\3D Objects>cd tc23

C:\Users\PRASHANT\3D Objects\tc23>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 677-589-813
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/May/2023 11:09:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/download.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/ramayan.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/ rrr.jpg HTTP/1.1" 304 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /static/images/video.mp4 HTTP/1.1" 206 -
127.0.0.1 - - [19/May/2023 11:09:19] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [19/May/2023 11:09:39] "POST /classify HTTP/1.1" 200 -
```



```
index.html X
C:\Users\PRASHANT>3D Objects>tc23>templates>index.html>...
115 <video id="video-background" autoplay muted loop>
116 <source src="{ { url_for('static', filename='images/video.mp4') } }" type="video/mp4">
117 </video>
118 <h1>Review Classification</h1>
119 <div class="image-grid">
120 <div class="image-input-container">
121 <div class="image-container">
122 
123 <div class="image-name">RRR</div>
124 </div>
125 <div class="classification-form">
126 <label for="input-text1">Enter Review:</label>
127 <textarea id="input-text1" placeholder="Enter your text here..." required></textarea>
128 <button type="button" onclick="classify('input-text1', 'result1')>Classify</button>
129 <div id="result1" class="result-box"></div>
130 </div>
131 </div>
132 <div class="image-input-container">
133 <div class="image-container">
134 
135 <div class="image-name">Elephant Whisperers</div>
136 </div>
137 <div class="classification-form">
138 <label for="input-text2">Enter Review:</label>
139 <textarea id="input-text2" placeholder="Enter your text here..." required></textarea>
140 <button type="button" onclick="classify('input-text2', 'result2')>Classify</button>
141 <div id="result2" class="result-box"></div>
142 </div>
143 </div>
144 <div class="image-input-container">
145 <div class="image-container">
146 
147 <div class="image-name">Ramayan</div>
148 </div>
149 <div class="classification-form">
150 <label for="input-text3">Enter Review:</label>
151 <textarea id="input-text3" placeholder="Enter your text here..." required></textarea>
152 <button type="button" onclick="classify('input-text3', 'result3')>Classify</button>
```

## Database:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 50000 entries, 0 to 49999  
Data columns (total 2 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   review      50000 non-null  object  
1   sentiment   50000 non-null  object  
dtypes: object(2)  
memory usage: 781.4+ KB
```

```
df=pd.read_csv('IMDB Dataset.csv')  
df.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production.   The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

## **Conclusion**

In conclusion, based on our analysis of the movie review text classification, the model exhibits a commendable ability to categorize reviews accurately.

However, it should be considered as a tool to assist in the classification process rather than a definitive judgment.

With further refinement, the model has the potential to enhance the efficiency and effectiveness of sentiment analysis in movie reviews, thereby providing valuable insights to filmmakers, critics, and audiences alike.

## References

1. <https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>
2. [imgurl:http://127.0.0.1:5000/static/images/m.jpg](http://imgurl:http://127.0.0.1:5000/static/images/m.jpg) - Bing
3. [Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK. | by Javed Shaikh | Towards Data Science](#)