**Mini Project Report on**



**TEXT CLASSIFICATION USING PYTHON**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

**RUDRAKSH AGARWAL 2021907**

***Under the Mentorship of***

**Dr. Surender Singh Samant**

**Associate Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“ TEXT CLASSIFICATION USING PYTHON ”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Surender Singh Samant, Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name University Roll No.

RUDRAKSH AGARWAL 2021907

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-4** |
| Chapter 3 | Methodology | **5-7** |
| Chapter 4 | Result and Discussion | **8-9** |
| Chapter 5 | Conclusion and Future Work | **10** |
|  | References | **11** |

**Chapter 1**

**Introduction**

The Purpose of this project is to develop a system for Text Classification using Python Libraries. The System should be able to predict the category of the given text accurately.

* 1. **Introduction**

Text Classification is a machine learning technique which specifies a set of predefined categories to a given text. Here, in this project we have focused on implementing text classification using python, a general high level programming language with powerful libraries like Scikit learn, Numpy, Matplotlib, etc. Text Classification has many applications, some of them are:

**1.1.1 Spam Detection**

The word spam refers to the unsolicited or irrelevant messages send over the internet, with the purpose of advertising, spreading malware or other unwanted activities. Spam Detection is a procedure to identify those irrelevant messages, such messages are usually sent through electronic communication such as emails & SMS.

**1.1.2 Sentiment Analysis**

The Word Sentiment refers to the emotions of an individual or a group of people. Sentiment Analysis is used for identifying sentiment tone in a given text. It is generally used for getting public opinion & trends of various domains. Some applications of Sentiment Analysis are: Product Review, Customer Feedback & Brand Monitoring.

**1.1.3 Fake News Detection**

Fake News Detection is the process which generally defines the genuineness of a given news text. It helps in preventing that audience getting manipulated with false claims and fake news.

Fake News Detection includes content analysis, source verification, cross checking, social media analysis, public opinion analysis.

**1.1.4 E-commerce Product Classification**

E-commerce or Electronic commerce is playing a vital role in managing products of different categories and of different brands in multiple classifications according to their type, price, offer, discount, etc. E-Commerce Product Classification is helpful in improving customers shopping experience, increased sales, more accurate search results.

**1.1.5 Legal Document Classification**

Legal Documents can be classified on the basis of their content and purpose. Legal Document Classification is helps in Efficient Retrieval, it saves time and makes the process cost efficient, helps in easy collaborations. Legal Document Classification includes Document type, subject, jurisdiction, date and time, status of confidentiality, references.

**1.2 Objective**

The Main Objectives of Text Classification using python are:

1. To design & develop a text classification model using Python.
2. To Evaluate the performance of the model on a given dataset.
3. To analyze and describe the outputs of the classification.

**Chapter 2**

**Literature Survey**

A literature Survey on Text Classification using Python includes research on various methods used to accurately classify the text from the predefined categories. Some Previous Studies stated that text classification has a vast scope in various aspects. Some of them are:

In a previous case study by David Colton [1], he had used the Natural Language Toolkit and scikit-Learn library on multiple datasets by modelling with both the methods and develop NLTK Naïve bayes classifier, Naïve bayes scikit-learn classifier and Support vector machine scikit-learn classifier. He had concluded that binary text classification using python, NLTK, scikit-learn results that scikit-learn SVM Classifier provides slightly better results than NLTK & scikit-learn Naïve Bayes Classifier.

The Research Conducted by Hovy D. and published online by Cambridge University in 2022 [2] focused on the Prediction and Classification for Social Scientists in which they concluded that text classification provides the social scientists with the most common method of text classification. They have also stated that prediction algorithms have become more powerful, especially for neural network methods.

The Research Conducted by Shachi Shula in 2018 [3] focused on sentiment analysis to extract the emotions and public opinion for a particular topic from the given set of data set. Here, they have considered IMDB movie review database. They have concluded that the improved Naïve bayes classifier with TF-IDF (Term Frequency-Inverse Document Frequency) shows better accuracy than any other classification algorithm.

In a research paper, published by Bijoyan Das & Sarit Chakraborty [4], they proposed that text classification can be done using Term Frequency-Inverse Document Frequency (TF-IDF) along with Next Word Negation. They have applied their TF-IFD & TF-IDF-NWN models on multiple algorithms, From Which they concluded that Linear Support Vector Machine is the most suitable algorithm.

The Research Paper published by IEEE, in 2019 Open Conference of electrical, electronic and information sciences(estream) [5], they have considered three different approaches for text classification including, TF-IDF, TF-IDF with Latent Semantic Analysis (LSA) & TF-IDF with Linear Discriminant Analysis (LDA), out of which they concluded with the results that the TF-IDF gives the highest accuracy even with large datasets.

**Chapter 3**

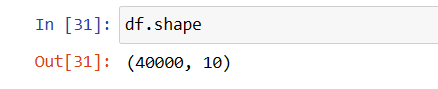
**Methodology**

**3.1 Dataset**

Here, we have used the Amazon Product Review Dataset in which we have customer reviews for multiple product categories for which we have to train the model in such a way, so that, it can give the accurate category as the result for any given text.

**3.1.1 Shape of Dataset**

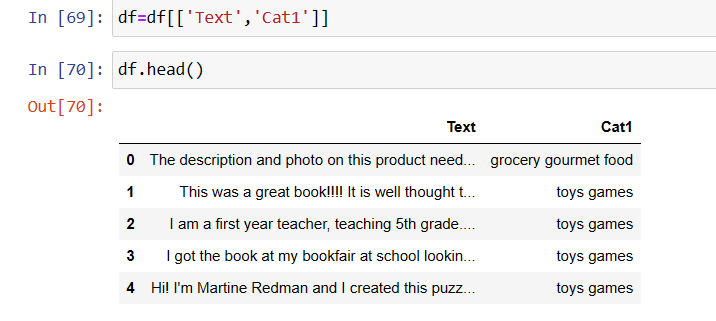
Here, we have considered a large dataset of the shape as shown in Fig. 3.1

****

**Fig. 3.1**

**3.1.2 Selection for Text Columns**

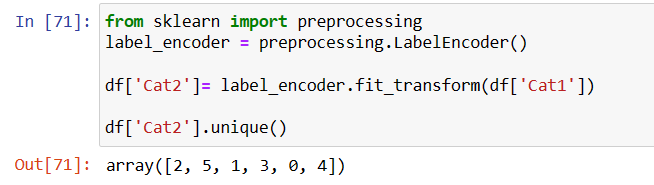
In such a large dataset, we have considered only two text columns as shown in Fig.3.2, i.e., Text column & Category column.



**Fig 3.2**

**3.2 Preprocessing**

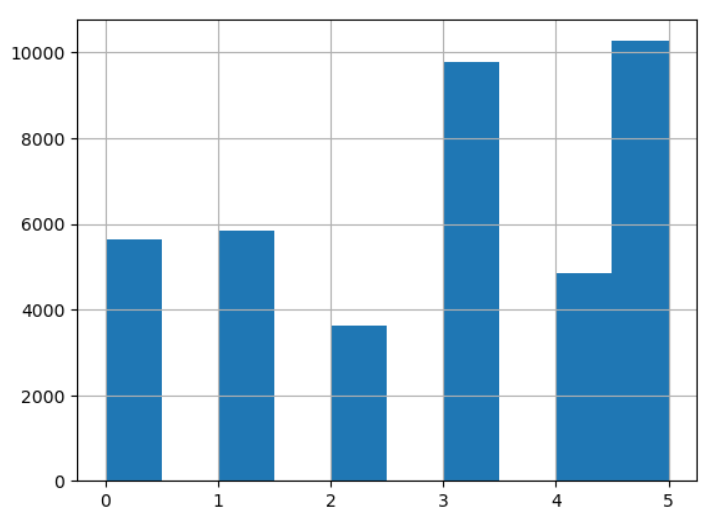
Now, initially we have to clean & preprocess the dataset to make it more suitable for the model. For which we have firstly encode all possible categories using label encoder, which is a method of the preprocessing module of the scikit-learn library, as shown in Fig. 3.3. Here we have created a new category say Cat2 in which we’ll store those unique labels assigned for each category. Also we have checked for the null values from the dataset which helps in check the discrepancies of the dataset.



**Fig. 3.3**

**3.2.1 Category Distribution**

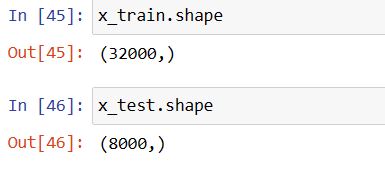
Here, we have displayed the dataset distribution among all six categories using a histogram as shown in Fig. 3.4.

****

**Fig. 3.4**

**3.3 Data Splitting**

We have made a dataset split between testing data and training data, i.e., 80% Training Dataset & 20% Testing Dataset, as shown in Fig. 3.5.



**Fig. 3.5**

**3.4 Model Selection**

Now, we have selected the Count Vectorizer of feature extraction module of sklearn library as it is one of the most popular Natural Language Processing Technique for text classification. It is also well suited for traditional machine learning models.

The Multinomial Naïve Bayes Classifier, for training and checking the test predictions, is used as it is one of the most popular Naïve Bayes especially for Text Classification because of it efficiency and effectiveness.

**3.5 Training & Evaluation**

We have to train our model for the training dataset using the Multinomial Naïve Bayes Classifier, after that we will check the prediction for the testing dataset. Later, we have to prepare its classification report, including precision, recall. F1-score, Support along with accuracy score, using the classification\_report as shown in Fig. 3.6 and accuracy\_score module of the sklearn.metrices library as shown in Fig. 3.7.

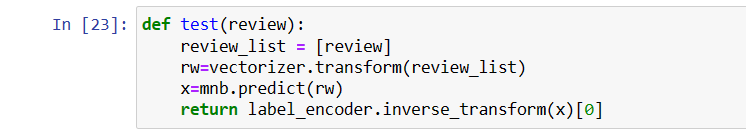
****

**Fig. 3.6**

****

**Fig. 3.7**

At the end we have used a test case for taking reviews as input and give the desirable output as shown in Fig. 3.8.

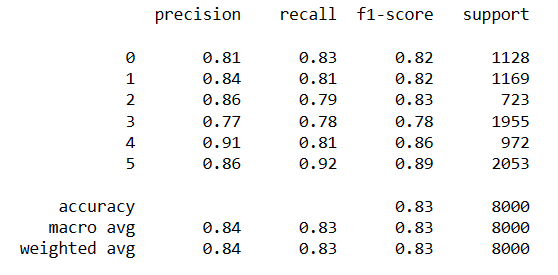
****

**Fig. 3.8**

**Chapter 4**

**Result and Discussion**

The Results for the Text Classification Model are: The Classification report, including precision, recall, F1-score, support, are as shown in Fig. 4.1.

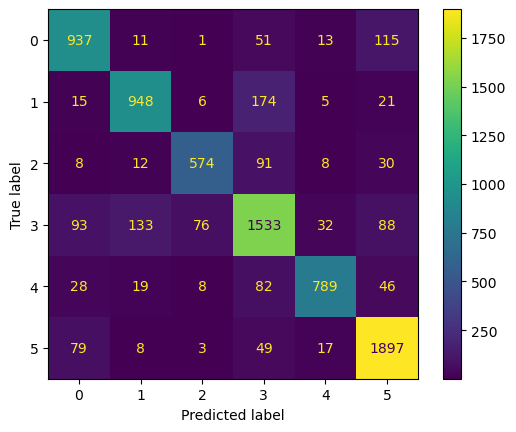


**Fig 4.1**

The accuracy score of the model for Text Classification is as shown in Fig. 4.2 :

**Fig 4.2**

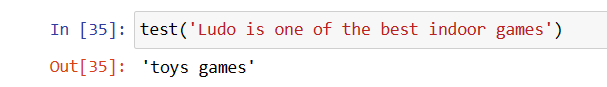
Now, we have displayed a confusion matrix for True Label and Predicted Label, which shows the number of True Predictions after comparing the predefined True Labels with the Predicted Labels as shown in Fig 4.3



**Fig, 4.3**

Here, for this Trained model we have multiple test cases for each predefined category of products for different customer reviews as shown in Fig. 4.4.

**4.1 Toys Games**

****

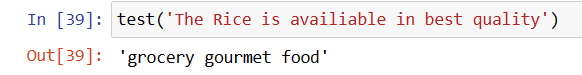
**Fig. 4.4(a)**

**4.2 Beauty Products**

****

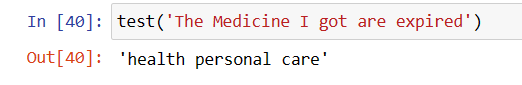
**Fig. 4.4(b)**

**4.3 Grocery & Gourmet Foods**

****

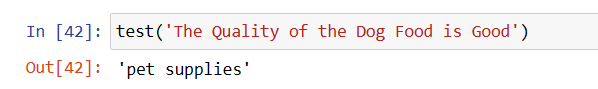
**Fig. 4.4(c)**

**4.4 Health & Personal Care**

****

**Fig. 4.4(d)**

**4.5 Pet Supplies**

****

**Fig. 4.4(e)**

**4.6 Baby Products**

****

**Fig. 4.4(f)**

**Chapter 5**

**Conclusion and Future Work**

**5.1 Conclusion**

The Text Classification model developed in this project using python is able to accurately predict the desired category from the predefined categories for the given text input. The model’s highest accuracy rate is suitable for use in a variety of applications like Hate Speech detection, Email/SMS spam Detection, E-commerce Product Categorization, Sentiment Analysis, Legal Document Classification, etc.

Text Classification is a very useful tool for many more applications like Language Detection, Product Recommendations in E-Commerce Apps like Amazon & Flipkart, Social Media Monitoring. The use of Python & Machine Learning Libraries makes it easier for the model to get easy and more appropriate predictions for the predefined true labels. In this Model, we have added multiple categories for the different types of texts in the dataset. We have also checked the performance of the model using the classification report, accuracy score and confusion matrix display.

**5.2 Future Work**

In The Field of Text Classification, till now we have so much advancements with multiple ways and different algorithms to perform this task, but still, we require certain advancements, so that, can improve the user experience and resolves the code complexity. We should also add certain personalization features in the model so that the user can classify whatever he/she wants. We have to develop the domain adaption and adaptive learning feature, so that, the model can automatically predict the correct domain for any given textual data. The Addition of both textual and non-textual features is also helpful in creating a Text Classification model with Multimodal Data Handling ability. We should also increase the Maximum number of domains to predict using the large datasets with a diverse range of topics, languages and domains.

**References**

[1] Hofmann, M., & Chisholm, A. (2016). Text mining and Visualization: Case Studies Using Open Source Tools. CRC Press.

[2] Hovy D. Prediction and Classification in Text Analysis with Python for Social Scientists. 1017/9781108960885. Cambridge, MA: Cambridge University Press, 2022

[3] An Improved Machine Learning Approach to Analyze the Sentiment of the Movie Reviews Using IMDB dataset Shachi Shukla22 December 2018The International Institute for Science. (n.d.). IISTE

[4] Das, B., & Chakraborty, S. (2018). An improved text sentiment classification model using TF-IDF and Next Word Negation. In arXiv [cs.CL]. <http://arxiv.org/abs/1806.06407>

[5] R. Dzisiewicz and D. Szeshok, “Text classification using different feature extraction approaches,” 2019 Open Conference on Electrical, Electronic and Computer Sciences (eStream), Vilnius, Lithuania, 2019, pp. 1–4. doi: 10.1109/eStream. 2019.8732167.