**Social Media Sentiment Analysis**

A project report

By

*Ashwin Shetty*

*Computer and Communications Engineering*

*Manipal Institute of Technology*

*Manipal*

**Abstract**

Sentiment analysis is a powerful tool in natural language processing (NLP) that involves finding the emotion a user wished to convey with their text. This is extremely useful in fields such as market analysis, customer feedback monitoring, and social media analytics. This project uses a social media dataset, and applies multiple preprocessing techniques to help the model predict more accurately. A bidirectional LSTM is used for sentiment analysis.

**Acknowledgements**

Success of any project depends largely on the encouragement and guidelines of many others. I take this sincere opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project work.

Words are inadequate in offering my thanks to the other trainees, project assistants and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

**TABLE OF CONTENTS:**

* Overview
* Python
* Artificial Intelligence and Machine Learning
* RNN and LSTM
* Bidirectional Layers
* Social media sentiment analysis
  + Imports
  + Dataset
  + Preprocessing
  + Model
  + Validation
* Conclusion

**OVERVIEW**

This project aims to create a sentiment analysis tool for classifying text data from sources like social media posts and reviews. The system processes text to identify sentiments and label them as positive, neutral, or negative.

**Key Steps:**

1. **Data Preparation**:
   * **Cleaning**: Remove punctuation, convert text to lowercase, and eliminate stop words.
   * **Normalization**: Apply lemmatization and convert emojis to text descriptions.
2. **Handling Imbalance**:
   * Use oversampling techniques to balance the dataset, ensuring equal representation of sentiment categories.
3. **Model Development**:
   * Train and evaluate machine learning and deep learning models like Bidirectional LSTM.
4. **Visualization**:
   * Create confusion matrices and precision-recall curves to analyse model performance.

**Goals:**

* **Accuracy**: High precision in sentiment classification.
* **Scalability**: Handle large datasets efficiently.
* **Versatility**: Apply to various fields like market analysis and customer feedback.

**PYTHON**

Python is a versatile and widely used high-level programming language known for its readability, simplicity, and ease of use. Created by Guido van Rossum and first released in 1991, Python has grown to become one of the most popular programming languages in the world.

**Key Features:**

1. **Readability and Simplicity**:
   * Python's syntax is clean and straightforward, making it accessible for beginners and allowing developers to write code more efficiently.
2. **Dynamic Typing**:
   * Variables in Python do not need explicit declaration, which speeds up development and reduces boilerplate code.
3. **Interpreted Language**:
   * Python is an interpreted language, meaning code is executed line by line, which simplifies debugging and testing.
4. **Extensive Standard Library**:
   * Python comes with a vast standard library that supports many common programming tasks, from file handling and networking to web development and data analysis.
5. **Cross-Platform Compatibility**:
   * Python is available on multiple platforms (Windows, macOS, Linux), making it a versatile choice for developers.
6. **Support for Multiple Programming Paradigms**:
   * Python supports procedural, object-oriented, and functional programming styles, providing flexibility in how developers approach problem-solving.

**Applications of Python:**

1. **Web Development**:
   * Frameworks like Django and Flask make it easy to build and deploy web applications.
2. **Data Science and Machine Learning**:
   * Libraries such as Pandas, NumPy, and Scikit-Learn, along with frameworks like TensorFlow and PyTorch, make Python a go-to language for data analysis and machine learning.
3. **Automation and Scripting**:
   * Python is often used for writing scripts to automate repetitive tasks and for system administration.
4. **Software Development**:
   * Python is used in developing standalone applications and has tools like PyInstaller to convert scripts into executables.

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**Artificial Intelligence (AI)**

**Definition**: AI is a broad field of computer science focused on creating systems capable of performing tasks that typically require human intelligence. This includes reasoning, learning, problem-solving, perception, and language understanding.

**Key Areas of AI**:

1. **Narrow AI**: Also known as Weak AI, it is designed and trained for a specific task, such as virtual assistants (e.g., Siri, Alexa), recommendation systems, and image recognition.
2. **General AI**: Also known as Strong AI, it is a theoretical form of AI that possesses the ability to understand, learn, and apply intelligence across a wide range of tasks at a level comparable to humans.
3. **Superintelligent AI**: A future concept where AI surpasses human intelligence in all aspects, including creativity, problem-solving, and social intelligence.

**Applications of AI**:

* **Natural Language Processing (NLP)**: Enables computers to understand, interpret, and generate human language. Examples include chatbots, language translation, and sentiment analysis.
* **Computer Vision**: Allows computers to interpret and make decisions based on visual data from the world, such as in facial recognition, autonomous vehicles, and medical imaging.
* **Expert Systems**: AI systems that emulate the decision-making ability of a human expert, used in areas like medical diagnosis, financial planning, and technical support.

**Machine Learning (ML)**

**Definition**: ML is a subset of AI that focuses on the development of algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data.

**Types of Machine Learning**:

1. **Supervised Learning**: The model is trained on a labelled dataset, where the input and the corresponding output are provided. Examples include regression and classification tasks.
   * **Example**: Predicting house prices based on features like size, location, and amenities.
2. **Unsupervised Learning**: The model is trained on an unlabelled dataset, and it must find patterns and relationships in the data. Examples include clustering and association tasks.
   * **Example**: Customer segmentation based on purchasing behaviour.
3. **Reinforcement Learning**: The model learns by interacting with an environment and receiving rewards or penalties for actions taken. This approach is often used in robotics and game playing.
   * **Example**: Training a robot to navigate a maze.

**Common Algorithms**:

* **Linear Regression**: Predicts a continuous target variable based on the linear relationship between input features.
* **Decision Trees**: Splits data into branches to make decisions and predictions.
* **Neural Networks**: Inspired by the human brain, these are used for complex pattern recognition tasks.
* **K-Means Clustering**: An unsupervised learning algorithm that groups data into clusters based on similarity.

**Applications of ML**:

* **Healthcare**: Predicting disease outbreaks, personalized treatment plans, and medical imaging analysis.
* **Finance**: Fraud detection, algorithmic trading, and credit scoring.
* **Marketing**: Customer segmentation, recommendation systems, and sentiment analysis.
* **Transportation**: Autonomous driving, traffic prediction, and route optimization.

**RNN AND LSTM**

**RNN**

RNN stands for Recurrent Neural networks. These are considered the best for processing text related sequences. RNN’s process sequences by iterating along the sequence elements and keeping information relative to what it has processed so far

**Key Characteristics:**

1. **Sequence Handling**:
   * RNNs are well-suited for tasks where the order of the data is important. They maintain a memory of previous inputs in the sequence, making them effective for sequential data.
2. **Memory Persistence**:
   * The looping connections in RNNs create an internal state that can capture information about previous steps in the sequence, which helps in predicting future steps.

**LSTM**

LSTM stands for Long Short term Memory. It is a type of RNN architecture specifically designed to capture long-term dependencies in sequence data. Introduced by Hochreiter and Schmid Huber in 1997, LSTMs are particularly effective at mitigating the vanishing gradient problem, which hampers the learning of long-term dependencies in standard RNNs.

**Key Characteristics:**

1. **Memory Cells**:
   * LSTMs consist of memory cells that can maintain information over long sequences. These cells are regulated by gates that control the flow of information.
2. **Gating Mechanisms**:
   * **Input Gate**: Controls the extent to which new information flows into the memory cell.
   * **Forget Gate**: Decides what information from the cell state should be discarded or retained.
   * **Output Gate**: Determines the output based on the cell state.

**Advantages:**

* **Long-Term Dependencies**: LSTMs effectively capture long-term dependencies in data, making them suitable for tasks requiring memory of previous information.
* **Gradient Flow**: The gating mechanisms help maintain a constant error flow, addressing the vanishing gradient problem.

**BIDIRECTIONAL LAYERS**

Bidirectional layers model consists of two RNNs (LSTMs or GRUs) that process the input sequence in one different direction to finally merge representations. By doing this, they’re able to catch more complex patterns than a single RNN layer would catch.

One RNN processes the input sequence in a forward direction (from start to end), while the other processes it in a backward direction (from end to start)

**Applications:**

* **Natural Language Processing (NLP)**: Sentiment analysis, machine translation, named entity recognition, and text classification.
* **Speech Recognition**: Improving the accuracy of converting speech to text by leveraging future context.
* **Time Series Prediction**: Better predictions by understanding trends and patterns in both directions.

**SOCIAL MEDIA SENTIMENT ANALYSIS**

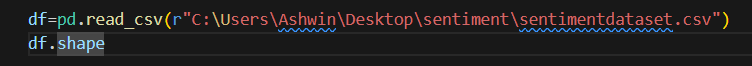
**IMPORTS**

Basic imports

A screenshot of a computer screen

Description automatically generated

Dataset



For plotting graphs

A black background with colorful text

Description automatically generated

For text preprocessing and creating training-testing set

A screen shot of a computer program

Description automatically generated

For the ML model

A screen shot of a computer

Description automatically generated

**DATASET**

This project utilizes the social media analysis dataset from

https://www.kaggle.com/datasets/kashishparmar02/social-media-sentiments-analysis-dataset/data

This dataset contains 732 rows and 15 columns. It contains 279 different emotions, such as Positive, Happiness, Joy,  Fear, Sadness etc.

A screen shot of a black and white screen

Description automatically generated

A graph with blue bars

Description automatically generated

A computer screen shot of a code

Description automatically generated

From this we can analyse the following:

Dataset contains texts which might have both lower and uppercase letters, emojis etc.

The number of emoji’s is only 11, so it’s not large enough to affect the model

The number of positive texts exceed the number of negative texts

These must be addressed during preprocessing.

**PREPROCESSING**

A screen shot of a computer program

Description automatically generated

We first convert emojis to their textual description. We then remove punctuations and convert all text to lowercase.

We then tokenise the text and then remove stop words

Stop words are words such as ‘I’, ‘are’, ‘is’ etc. which do not affect the sentiment shown by the sentence. They may hinder the ML model and are hence removed.

Lemmatizers are used to convert some words to their root form for better understanding by the model.

A screenshot of a computer

Description automatically generated

We then need to generalize all 279 emotions into either positive, neutral or negative.

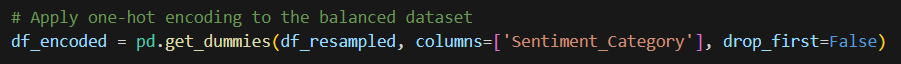
We came across the fact that the number of positive texts exceed neutral or negative. Thus we need to make duplicate samples for negative and neutral to bring them all to a closer value

A computer screen with colorful text

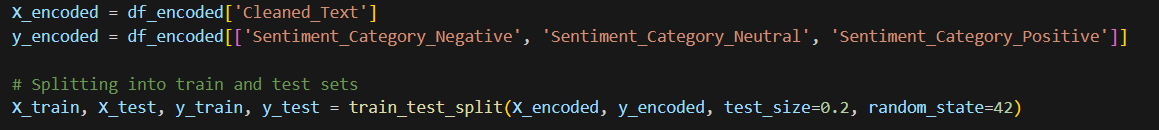
Description automatically generated

We utilize Random Oversampler for this.

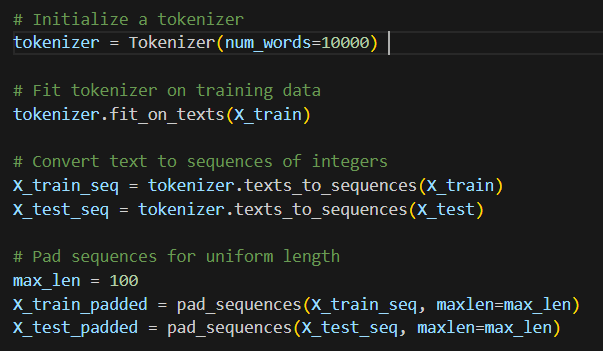
We now apply one hot encoding to Sentiments. This means we split sentiment into Positive, Negative and Neutral columns and assign 1 or 0 to each column depending on the sentiment of the text.



We can finally split into a train test split



ML model cannot understand strings. So we apply a tokenizer to convert them to a sequence of integers



The strings will now look like this:

A black background with white numbers

Description automatically generated

**MODEL**

We can now put this into our bidirectional LSTM layers RNN model.A black background with colorful text

Description automatically generated

This sequential model consists of 3 main layers

The first layer takes the input.

The second layer consists of a bidirectional layer, specifically two LSTM layers.

The third layer is a normal dense layer with softmax activation function, which is used for multiclass classification problems.

**VALIDATION**

****

This model shows a accuracy of 99.89% on the training set and 89.19% on the testing set.

A blue squares with white text

Description automatically generated

We can see that it predicts most of the sentiments correctly. In total, it predicted 24 out of 222 texts wrong. Largely the errors occurred at differentiating neutral and negative texts.

A graph with different colored lines

Description automatically generated

**CONCLUSION**

This model gave a 89.19% accuracy for test data, which is a good accuracy. Changing parameters gave less accuracy so this model was chosen as the best model. Some of the improvements possible are:

-Larger dataset

-better utilization of emojis to predict sentiments