**REPORT**

**ON**

**SENTIMENT ANALYSIS IMPLEMENTATION USING VARIOUS**

**MACHINE LEARNING MODELS**

Nikhil Dugyala- G25443567

Hemanth Kumar Reddy Kasireddy- G26247387

Apoorva Mamindla-G38565019

**Problem Definition:**

The goal of sentiment analysis on the IMDb dataset would be to develop a model that can accurately classify movie reviews as either positive or negative, which can have a variety of real-world applications, such as improving product recommendations, understanding customer feedback, and identifying trends in public opinion.

**Motivation**: The motivation for sentiment analysis is to leverage the power of natural language processing and machine learning to gain insights from large amounts of text data, which can help us make better decisions and improve the quality of products and services.

**Hypothesis**: The implementation of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models can improve the accuracy of sentiment analysis on movie reviews compared to traditional machine learning models such as logistic regression or SVM.

**Assumptions**:

* The dataset is representative: The dataset needs to cover a wide range of examples to avoid bias towards certain types of data.
* The model architecture is appropriate: The model architecture should be suitable for the problem at hand. If it's too complex or too simple, it may not perform well.
* The hyperparameters are tuned appropriately: The learning rate, batch size, and number of epochs should be optimized for the dataset and model architecture to achieve the best performance.

**Approach**: For our sentiment analysis project, we have decided to use a deep learning approach, specifically a convolutional neural network (CNN) and a long short-term memory (LSTM) model. We chose this approach because deep learning models have shown state-of-the-art performance in many NLP tasks, including sentiment analysis.

**Existing Methods**: Other methods we considered for sentiment analysis include traditional machine learning algorithms such as logistic regression, XG Boost, Naive Bayes and SVM. However, we chose deep learning models because they have shown superior performance in many NLP tasks, especially in cases where the task involves complex relationships and dependencies between words in the text.

**Is this approach Novel?**

The approach of using deep learning models such as CNN and LSTM for sentiment analysis is not a novel approach. However, our proposed approach may differ from existing approaches in the specific implementation details, such as the architecture of the models, the preprocessing steps, and the hyperparameter tuning process.

**Data Collection and Preprocessing**

**Dataset:**

The dataset which we will use in sentiment analysis is the International Movie Database (IMDb) reviews for 50,000 reviews of movies from all over the world, its a binary classification dataset categorizing each review in a positive or negative. It has 25000 samples for training and 25000 for testing.

Dataset Link: <http://ai.stanford.edu/~amaas/data/sentiment/>

**Exploratory data analysis (EDA) and data visualization:**

**A picture containing text, screenshot, font, number

Description automatically generated** Graphical user interface, application

Description automatically generated with medium confidence

Chart, histogram

Description automatically generatedA picture containing text, font, white

Description automatically generated

There are 25,000 reviews labeled as positive and 25,000 reviews labeled as negative.

Chart, bar chart, treemap chart

Description automatically generated

We have analyzed the most common words in the positive and negative reviews of the dataset.

Chart, bar chart, histogram

Description automatically generatedChart, bar chart

Description automatically generated

**Data cleaning and preprocessing steps:**

The purpose of the preprocessing is to clean and prepare the data for use in machine learning models that can classify the reviews as positive or negative. We have used multiple steps to preprocess the data.

1. We have checked for the missing values in the dataset.

A black text on a white background

Description automatically generated with low confidence

2. Converted reviews to lowercase and removed duplicate reviews, there were 418 duplicate reviews present in the dataset.

3. We have cleaned the reviews by removing any HTML tags, square brackets, noisy text, and special characters.

4. Lemmatized the Text, converting a word to its base form or lemma. For example, the lemma of the word 'running' is 'run' and Stop words such as 'the', 'a', 'an', 'in', 'and', etc. are removed from the reviews.

**Feature Extraction and Representation:**

We are implementing this project using the two feature extraction methods namely Bag of Words, TF-IDF methods.

**MACHINE LEARNING MODELS IMPLEMENTATION:**

We have built and experimented with different ML models to compare their performance on the dataset.

**LOGISTIC REGRESSION**:

The hyperparameters of the logistic regression model are penalty='l2', C=1, and random state=23. The performance evaluation metrics used to evaluate the logistic regression model are accuracy, precision, recall, and F1-score. The accuracy score for the logistic regression model using BOW features is 0.739, and the accuracy score for the logistic regression model using TF-IDF features is 0.734. The precision and recall values for both models are around 0.72-0.77, and the F1-score for both models is around 0.73.

A picture containing text, screenshot, font, number

Description automatically generatedChart, treemap chart

Description automatically generated

The confusion matrix shows that the model correctly classified 3999 positive sentiment reviews and 3259 negative sentiment reviews using TF-IDF features.

**Naive Bayes classifier**

Accuracy, precision, recall, and F1-score were used as performance evaluation metrics. The accuracy score of the Naive Bayes model using BOW features is 0.739 and using TFIDF features is 0.74. The precision and recall scores for both models are around 0.74, indicating that the model has good precision and recall for both positive and negative sentiments. The F1-score for both models is 0.73, indicating that the model has balanced performance between precision and recall.

A picture containing text, screenshot, font, number

Description automatically generatedChart, treemap chart

Description automatically generated

The model predicted 3784 positive sentiments and 3623 negative sentiments correctly for BOW features.

**SVM Model:** The implemented model is Support Vector Machines (SVM) with a linear kernel, which is used for binary classification. It is implemented using the Linear SVC class from the scikit-learn library in Python.

Model Selection Criteria:

The selection of SVM model is based on its ability to handle high dimensional data, good generalization performance, and robustness to outliers. The model selection criteria for SVM model include accuracy, precision, recall, and F1-score. The accuracy score of SVM using BOW and TF-IDF features are 0.738 and 0.739 respectively.

A picture containing text, screenshot, font, number

Description automatically generatedChart, treemap chart

Description automatically generated

**NEURAL NETWORKS IMPLEMENTATION**:

Data Preparation:

Overall, the Data Preparation step involves loading the dataset, converting the categorical sentiment labels to numerical values, and splitting the training data into actual training and validation data for use in training and evaluating the model. Our split is 65% training set, 15% Validation set and 20% testing set.

**Tokenization**: We have used the tokenization process to convert textual data into numerical form, which is required to feed data into a neural network.

**Padding**: Padding is performed on the sequences of numerical values generated from tokenization to ensure that all sequences have the same length, which is required for inputting the data into the neural network. In this code, the pad\_sequences function from the Keras preprocessing library is used to pad the sequences with zeros to a maximum length of 300.

**CONVOLUTIONAL NEURAL NETWORKS:**

**Model Architecture:**

**Table

Description automatically generated**

Our CNN model consists of five layers a Embedding layer, a 1D Convolutional layer, Max Pooling layer, a Global Max Pooling layer, and Two Dense layers.

The model is compiled using the compile() method of the Keras Sequential API. The Adam optimizer and the binary cross-entropy loss function is used. Our callback list includes ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, and TerminateOnNaN.

Accuracy is calculated as the fraction of correctly classified instances in the test set. Best Validation Accuracy = 89.70 and Best Training Accuracy = 99.85

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

In conclusion, we have successfully built a convolutional neural network model to classify movie reviews as positive or negative based on their text content. Our model achieved an accuracy of 89.31% on the test set, which indicates that it can accurately predict the sentiment of movie reviews.

A picture containing text, screenshot, diagram, font

Description automatically generatedA picture containing text, line, screenshot, plot

Description automatically generated

Confusion Matrix ROC Curve for this CNN model

**LSTM Deep Learning Model  
Table

Description automatically generated**

**Model Architecture:**

We then created the LSTM model using the Sequential() class from the tensorflow.keras module. The first layer of the model is an Embedding() layer. After the Embedding() layer, we added a LSTM() layer with 32 units. We added the 3 dense layers and a drop out layer.

Finally, we compiled the model using the compile() method. We used the binary\_crossentropy loss function and the adam optimizer, which is a popular optimization algorithm. Our callback list includes ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, and TerminateOnNaN to avoid the overfitting.

The model performed reasonably well on the given dataset, achieving a validation accuracy of 87.60% and Best Training Accuracy = 95.91. The training accuracy and validation accuracy are plotted against the number of epochs, as shown below:

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Testing the Final Model: We evaluated the performance of the trained model on the test dataset using the evaluate() method. We obtained the test accuracy and test loss as follows test Accuracy : 88.23

A picture containing text, screenshot, diagram, font

Description automatically generatedA picture containing text, line, screenshot, plot

Description automatically generated

Confusion Matrix ROC Curve for this LSTM model

Based on the results and classification report, the LSTM model achieved an overall accuracy of 88.2% on the test dataset, which is a good result. The f1-score for both classes is also above 0.85, which is considered good.

**Error Analysis:**

During the implementation of the sentiment analysis model, we encountered some errors and issues that required further investigation and troubleshooting.

* While training the deep learning models, we encountered an out of memory error due to the large size of the dataset and the complexity of the model architecture. To resolve this issue, we reduced the batch size and increased the number of epochs to compensate for the smaller batch size.
* During training, we noticed that some of the deep learning models were overfitting to the training data, resulting in poor generalization performance on the testing data. To address this issue, we introduced dropout regularization and early stopping to prevent overfitting.
* Despite achieving high accuracy scores on the testing data, we still observed some misclassification errors in the predictions made by the model. Upon further investigation, we found that some of the misclassification errors were due to subtle nuances and complexities in the language used in the movie reviews. To improve the model's performance, we experimented with different model architectures, hyperparameters, and preprocessing techniques.

**Future scope:**

* Explore the use of different pre-trained language models, such as GPT-3, BERT, or RoBERTa. These models have shown promising results in various natural language processing tasks and may improve the performance of sentiment analysis models.
* Performing sentiment analysis on text written in multiple languages. This can be achieved by either training separate models for each language or by using multilingual pre-trained language models.

**CONCLUSION:**

Based on the results presented, it can be concluded that the convolutional neural network (CNN) and long short-term memory (LSTM) models are better suited for sentiment analysis of movie reviews compared to the logistic regression, Naive Bayes, and SVM models. Both the CNN and LSTM models achieved high accuracy scores of 89.31% and 88.2%, respectively, on the test dataset. In addition, the precision, recall, and F1-score metrics were all above 0.85 for both models, indicating that they performed well in predicting both positive and negative sentiment. Baseline for sentimental analysis is around 80 to 85%, overall neural network approach performed better than other traditional ML baselines because they are highly flexible and can learn complex patterns in the data, making them suitable for tasks like sentiment analysis. The hyperparameters of the neural network, such as the learning rate, regularization strength, and number of hidden units, might have been tuned more effectively than in other baselines, leading to better performance.