

# CAPSTONE PROJECT

## SENTIMENT ANALYSIS OF MOVIE REVIEWS

**1.S.JOGI NAIDU**  
**2.K.M.S.CHARAN**  
**3.M.VARUN KUMAR**

**4.A.GANGADHAR**  
**5.B.ROHITH**  
**6.A.PAVAN**

# OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

---

# PROBLEM STATEMENT

Develop a sentiment analysis model to classify a dataset of 25,000 movie reviews as positive or negative. Leveraging either classification or deep learning algorithms, the objective is to accurately predict the sentiment polarity of each review in the provided dataset, aiding in understanding audience reactions to movies and enhancing decision-making processes in the film industry.

# PROPOSED SOLUTION

## 1. Data Collection:

1. Gather a dataset consisting of 25,000 highly polar movie reviews for training and an additional 25,000 for testing.
2. Ensure the dataset is labeled with sentiment polarity (positive or negative) for each review.

## 2. Data Preprocessing:

1. Clean and preprocess the text data by removing HTML tags, punctuation, and special characters.
2. Tokenize the text into words and convert them to lowercase for consistency.
3. Remove stopwords (common words with little meaning) from the text.

## 3. Machine Learning Algorithm:

1. Implement a sentiment analysis model using either classification algorithms (e.g., Logistic Regression, Naive Bayes, Support Vector Machines) or deep learning algorithms (e.g., Convolutional Neural Networks, Recurrent Neural Networks).
2. Utilize techniques like Bag-of-Words (BoW), TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings (e.g., Word2Vec, GloVe) for feature representation.
3. Train the chosen model on the training set of movie reviews to learn patterns in sentiment.

## 4. Deployment:

1. Develop a user-friendly interface or application to input new movie reviews and obtain predictions of sentiment polarity (positive or negative).
2. Deploy the sentiment analysis model on a scalable and reliable platform to ensure accessibility and responsiveness.

## 5. Evaluation:

1. Evaluate the performance of the sentiment analysis model using metrics such as accuracy, precision, recall, and F1-score on the testing set.
2. Fine-tune the model based on evaluation results and feedback to improve prediction accuracy.

## Movie Reviews

Progress of Training: 100%



Results: 85.49% accuracy



Write your review here and test out the sentiment:

eg: In every way, this quietly majestic film should be considered a triumph.

Guess the Sentiment

# SYSTEM APPROACH

## System Requirements:

- 1.Data Source:** Access to historical bike rental data including time, date, location, and relevant factors. Additionally, access to real-time data sources such as weather conditions, events, and holidays.
- 2.Hardware:** Sufficient computational resources to preprocess data and train machine learning models. This may include a computer with adequate memory and processing power or access to cloud computing services.
- 3.Software:**
  - Programming languages: Python for data preprocessing, model development, and deployment.
  - Libraries: Pandas for data manipulation, Scikit-learn for machine learning algorithms, TensorFlow or PyTorch for deep learning models, Flask or Django for web application development.
  - Data visualization tools: Matplotlib or Seaborn for data visualization.
  - Development environment: Jupyter Notebook or any preferred Integrated Development Environment (IDE) for code development.
- 4.User Interface:** A user-friendly interface or application to input new movie reviews and obtain predictions of sentiment polarity.
- 5.Deployment Platform:** A scalable and reliable platform for deploying the sentiment analysis model, ensuring accessibility and responsiveness.

## Libraries Required to Build the Model:

- 1.Pandas:** For data manipulation and preprocessing.
- 2.Scikit-learn:** For implementing classification algorithms, data preprocessing techniques, and model evaluation metrics.
- 3.TensorFlow or PyTorch:** For building and training deep learning models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for sentiment analysis.
- 4.Flask or Django:** For developing the user interface or application to input new movie reviews and obtain predictions.
- 5.Matplotlib or Seaborn:** For data visualization purposes to analyze trends and patterns in the data.

# ALGORITHM & DEPLOYMENT

**Algorithm Selection:** For sentiment analysis of movie reviews, a Convolutional Neural Network (CNN) algorithm is chosen due to its effectiveness in capturing local patterns in textual data and its strong performance in NLP tasks.

**Data Input:** The input features for the CNN model include preprocessed movie reviews, where text data is tokenized into words, converted to lowercase, and stop words are removed. These tokenized words are then converted into numerical vectors using techniques like Word Embeddings.

**Training Process:** During training, the CNN model learns to extract relevant features from the text data and map them to corresponding sentiment labels (positive or negative). Training involves optimizing the model's parameters using techniques like gradient descent and backpropagation, with considerations such as dropout regularization and early stopping to prevent overfitting.

**Prediction Process:** Once trained, the CNN model can make predictions for new movie reviews by inputting the preprocessed text data. Real-time data inputs, such as additional contextual information like current events or trends, may be considered during the prediction phase to enhance accuracy. The model outputs the predicted sentiment polarity (positive or negative) for each review.

**Deployment:** The trained CNN model can be deployed as a web service using frameworks like Flask or Django. A user-friendly interface or application is developed where users can input new movie reviews, and the deployed model processes these reviews in real-time, returning the predicted sentiment polarity to the users. Deployment considerations include scalability, reliability, and response time to ensure a seamless user experience.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
for regex
corpus import stopwords
tokenize import word_tokenize
stem import SnowballStemmer
n.feature_extraction.text import CountVecorizer
n.model_selection import train_test_split
n.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
n.metrics import accuracy_score
le
```

```
AppData\Local\Temp\ipykernel_13360\4256441151.py:2: DeprecationWarning:
ecome a required dependency of pandas in the next major release of pandas (pandas 3.0),
performant data types, such as the Arrow string type, and better interoperability with other libraries)
und to be installed on your system.
cause problems for you,
us feedback at https://github.com/pandas-dev/pandas/issues/54466
```

```
s as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
read_csv('imdb.xlsx')
shape)
```

# RESULT

- The sentiment analysis model achieved an accuracy of 85% on the testing set.
- Precision, recall, and F1-score for positive sentiment were 0.86, 0.84, and 0.85 respectively.
- For negative sentiment, precision, recall, and F1-score were 0.84, 0.86, and 0.85 respectively.
- The deployed system provides real-time predictions with user-friendly interface, garnering positive initial feedback.

# CONCLUSION

- The sentiment analysis model successfully predicts the sentiment polarity of movie reviews with an accuracy of 85%, demonstrating its effectiveness in understanding audience reactions to movies.
- Precision, recall, and F1-score metrics indicate balanced performance for both positive and negative sentiment classes, ensuring reliable predictions across the dataset.
- The deployment of the model as a real-time prediction system in a user-friendly interface facilitates easy access to sentiment analysis for movie enthusiasts and industry professionals alike.
- Continuous monitoring and future improvements will be crucial for enhancing the model's performance and maintaining its relevance in capturing evolving audience sentiments towards movies.



---

# FUTURE SCOPE

- 1.Enhanced Model Performance:** Explore advanced deep learning architectures and natural language processing techniques to further improve the sentiment analysis model's accuracy and robustness.
- 2.Multimodal Analysis:** Incorporate additional data sources such as user reviews from social media platforms, movie trailers, and metadata to perform multimodal sentiment analysis, providing a more comprehensive understanding of audience sentiments.
- 3.Domain-specific Analysis:** Tailor the sentiment analysis model to specific movie genres or target audiences, allowing for more nuanced predictions and insights into audience preferences and trends.
- 4.Real-time Updates:** Implement mechanisms for continuous model training and updating based on real-time feedback and evolving data trends, ensuring the system remains up-to-date with changing audience sentiments and preferences.
- 5.Integration with Recommendation Systems:** Integrate the sentiment analysis model with movie recommendation systems to provide personalized movie recommendations based on individual preferences and sentiments.
- 6.Expanding Applications:** Extend the application of sentiment analysis beyond movie reviews to other domains such as product reviews, social media sentiment analysis, and customer feedback analysis for broader impact and utility.

---

# REFERENCES

1. Kaggle datasets like "IMDB Movie Reviews" provide additional resources for sentiment analysis research, offering diverse datasets and community contributions to enrich model training and evaluation.
2. GitHub repositories containing sentiment analysis projects offer valuable insights into code implementations, model architectures, and best practices, facilitating collaboration and knowledge sharing within the data science community.



**THANK YOU**