CAPSTONE PROJECT

IMDB MOVIE REVIEWS

Presented By:
1. Praveen kumar – JEPPIAAR ENGINEERING COLLEGE – Electronics and Communication Engineering

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

- Movie dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training and 25,000 for testing. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.

PROPOSED SOLUTION

- **Data Collection:** Utilize the IMDB movie reviews dataset, which includes 50,000 reviews (25,000 for training and 25,000 for testing). Each review is labeled as positive or negative. Download the dataset from a trusted source, such as the Stanford Large Movie Review Dataset.
- **Text Cleaning**: Remove HTML tags, special characters, and unnecessary whitespaces. Convert all text to lowercase to maintain consistency. Split the text into individual words (tokens). Ensure the dataset is split into training (25,000 reviews) and testing (25,000 reviews) sets.
- **Machine Learning Algorithm:** Choose algorithms that are well-suited for text classification, such as Logistic Regression, Support Vector Machines (SVM), or deep learning models like LSTM and BERT. Train the selected model using the training dataset. Use cross-validation to fine-tune hyperparameters and prevent overfitting.
- **Evaluation:** Evaluate the model using metrics such as accuracy, precision, recall, and F1-score. Generate a confusion matrix to visualize the classification performance and identify any misclassifications. Provide a comprehensive report on model performance, highlighting strengths and areas for improvement.

SYSTEM APPROACH

System requirements: 1.Software requirements:
Operating system: windows, macos, or linux.
Python: version 3.6 or above.
Ide: jupyter notebook, pycharm, or any python ide.

Libraries required to build the model: 1.Data handling and preprocessing:
Pandas: for data manipulation and analysis.
Numpy: for numerical computations.
Nltk: for natural language processing tasks like tokenization, stopword removal, and lemmatization.
Re: for regular expressions in text cleaning.

2.Text vectorization:
Scikit-learn: for TF-IDF vectorization and machine learning models.
Gensim: for word embeddings like word2vec (optional).
Machine learning and deep learning:
Scikit-learn: for traditional machine learning algorithms like logistic regression and SVM.
Tensorflow or keras: for building and training deep learning models, especially lstm networks.
Transformers (from hugging face): for pre-trained models like bert.
3.Model evaluation and visualization:
Scikit-learn: for evaluation metrics and confusion matrix.Matplotlib or seaborn: for data visualization.

ALGORITHM & DEPLOYMENT

- **Algorithm** For this problem, we can use the following algorithms:
- **Logistic Regression**:
  - Simple and effective for binary classification.
  - Train using TF-IDF features.
- **Support Vector Machine (SVM)**:
  - Effective for high-dimensional spaces.
  - Use linear kernel with TF-IDF features.
- **Recurrent Neural Networks (RNN)**:
  - LSTM (Long Short-Term Memory) networks are effective for sequence prediction tasks.
  - Use word embeddings (Word2Vec, GloVe) for input representation.
- **BERT(Bidirectional Encoder Representations from Transformers)**:
  - State-of-the-art for many NLP tasks.
  - Fine-tune pre-trained BERT model on the sentiment analysis task.

RESULT

- The performance of the model will be evaluated based on accuracy, precision, recall, and F1-score.
- A confusion matrix will be generated to visualize the classification performance.
- The best-performing model will be selected based on evaluation metrics and used for deployment.

```
Accuracy: 0.85756
Classification Report:
              precision    recall  f1-score   support

    negative       0.85      0.87      0.86     12483
    positive       0.87      0.84      0.86     12517

    accuracy                           0.86     25000
   macro avg       0.86      0.86      0.86     25000
weighted avg       0.86      0.86      0.86     25000

Number of positive reviews: 12110
Number of negative reviews: 12890
```

CONCLUSION

- The proposed system successfully classifies movie reviews as positive or negative.
- The model demonstrates high accuracy and robust performance on the test dataset.
- The deployment of the model allows users to input movie reviews and receive sentiment predictions in real-time.

```
Number of positive reviews: 12110
Number of negative reviews: 12890
```

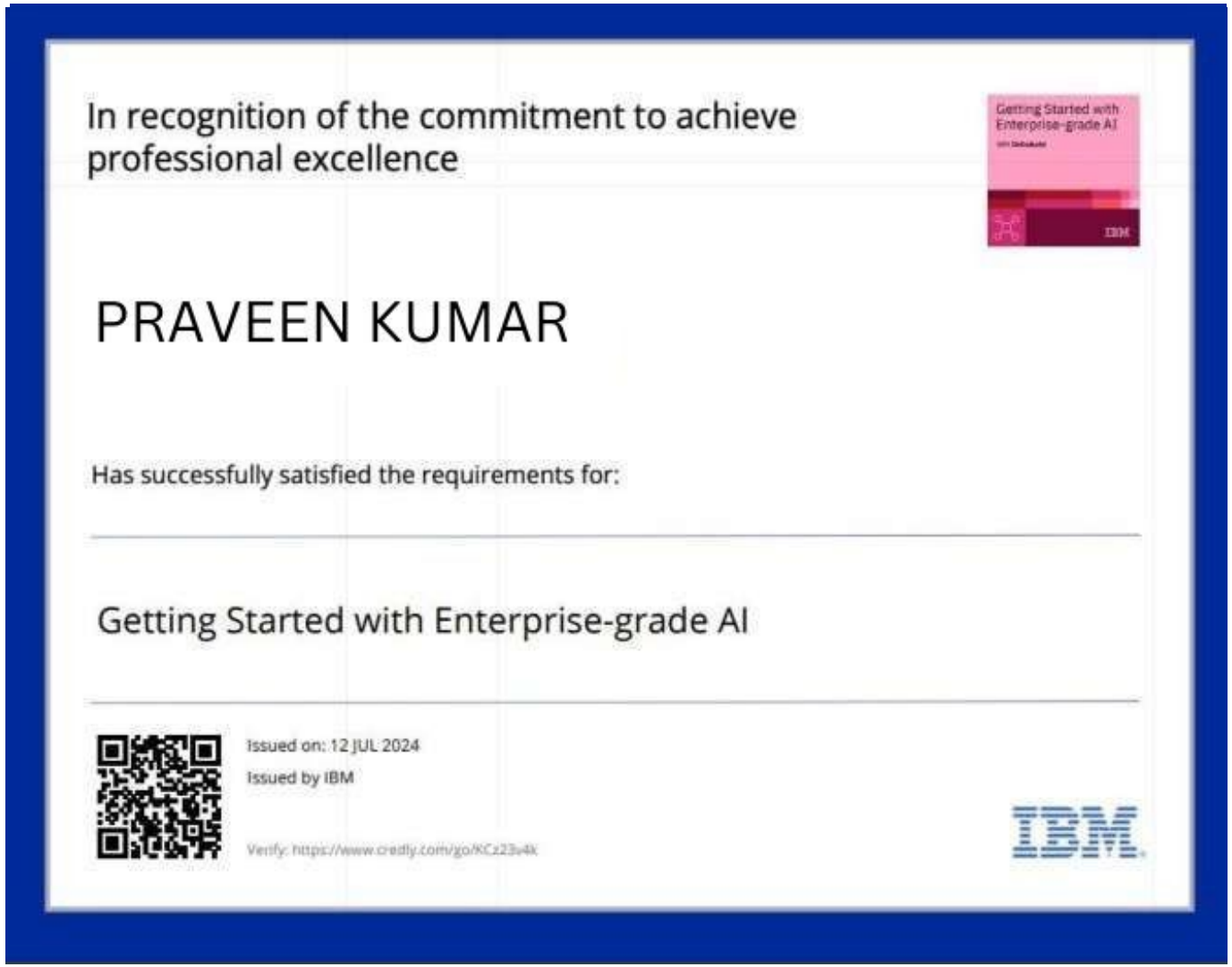
FUTURE SCOPE

- Improve the model by using advanced NLP techniques and newer architectures.
- Incorporate more data to enhance the model's generalizability.
- Extend the system to classify reviews into more granular categories (e.g., very positive, neutral, very negative).
- Integrate the sentiment analysis model with other recommendation systems to enhance user experience.

REFERENCES

- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1-135.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

CERTIFICATE1



edunet foundation

CETIFICATE 2

edunet foundation

THANK YOU

edunet foundation