# Fake-Review-Classification-and-Topic-Modeling

**# Objective:**

The goal of this project is to develop a system that can classify reviews as either fake or real by leveraging a combination of traditional machine learning, deep learning models, and Transformer-based approaches. Additionally, the system will cluster similar reviews together to group related feedback, making it easier to analyze customer insights. Furthermore, it will identify underlying topics within the reviews, providing a deeper understanding of customer sentiments and issues, ultimately helping to uncover patterns and concerns expressed by users.

**# Business Use Cases:**

Customer Trust: Detects fake reviews to protect customers from misleading info.

Product Feedback Analysis: Groups similar reviews to highlight common issues or features.

Content Moderation: Auto-filters fake or harmful reviews from product pages.

**# Step-by-Step Approach for the Project:-**

***1.Data Collection:***

Collect a Fake Review Dataset that includes product reviews along with metadata such as ratings, helpful votes, and other relevant information.

***2.Data Preprocessing:***

Text Cleaning: Remove special characters, numbers, and stop words from the reviews.

Tokenization: Break down the review text into individual words or tokens.

Text Vectorization: Convert the tokenized reviews into numerical representations using techniques like TF-IDF or Word2Vec.

***3.Topic Modeling (Unsupervised Learning):***

Latent Dirichlet Allocation (LDA): Apply this probabilistic model to assign each word in the reviews to one or more topics.

Non-Negative Matrix Factorization (NMF): Use NMF to factorize the word occurrence matrix and identify topics across all reviews.

***4.Clustering (Unsupervised Learning):***

K-Means Clustering: Group the reviews into K clusters based on similarities, such as product categories or review types.

DBSCAN: Implement DBSCAN, a density-based clustering algorithm that doesn’t require predefined clusters, to group reviews based on their content density.

**5.Fake Review Classification (Supervised Learning):**

**Traditional Machine Learning Models**: Use algorithms like Logistic Regression, Random Forest, and Support Vector Machine (SVM) to classify reviews as fake or real.

**Deep Learning Models:** Apply Long Short-Term Memory (LSTM) networks for sequential modeling and BERT (Bidirectional Encoder Representations from Transformers) for

context-aware text classification.

***6.Model Implementation:***

Tokenize the review texts and input them into the BERT or LSTM models for fake review classification.

Evaluate the performance of the models using metrics such as accuracy, precision, recall, and F1-score.

***7.Ensemble Methods:***

• Combine the results from different models to improve overall performance, such as:

• Combining predictions from both BERT and LSTM for a hybrid model approach.

• Using a weighted average of predictions from multiple classifiers to increase classification accuracy.

***8.Final evaluation:***

Measure performance using classification metrics such as accuracy, precision, recall, and F1 score.

**# Skills Takeaway from This Project:**

This project will enhance the skills in Python for programming, Pandas for data manipulation, and Scikit-Learn for machine learning tasks like classification and clustering.Also helps to gain experience with TensorFlow for deep learning models (LSTM, BERT) and Transformers (Hugging Face) for text classification.

**# Results and Conclusions from ML-Models**:

ML-model Accuracy Report

• Support Vector Machines (SVM): 87.02%

• K-Nearest Neighbors (KNN): 65.39%

• Decision Tree Model: 74.25%

• Random Forest Model: 84.85%

• Multinomial Naive Bayes: 84.36%

• Logistic Regression: 86.3%

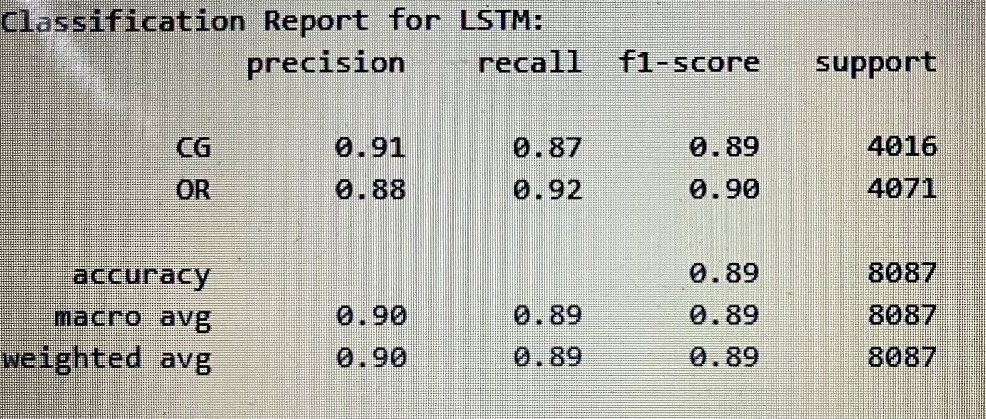
**Conclusion & Recommendations**

Best Performer: Support Vector Machine (SVM) achieved the highest accuracy, followed closely by Logistic Regression and Random Forest. These models should be considered as the primary candidates for deployment.

Underperforming Models: K-Nearest Neighbors (KNN) performed poorly with only 65.39% accuracy, suggesting that these models are not ideal for this dataset without further tuning or feature engineering

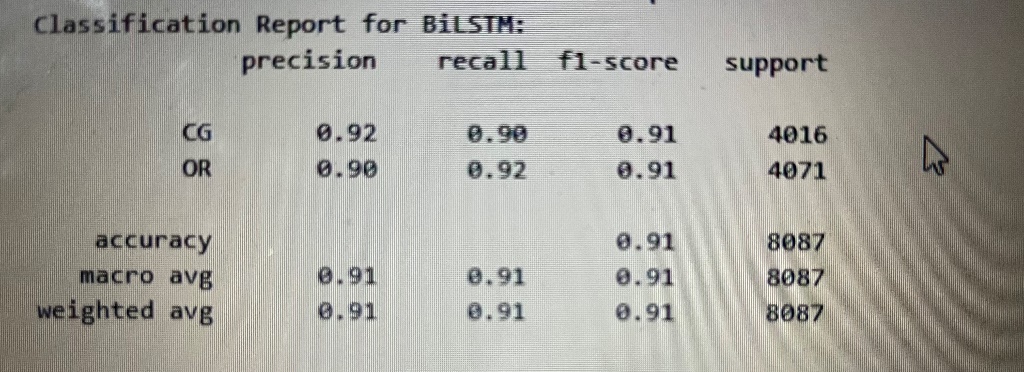
**# Results and Conclusions from DL-Models:**

**• Performance of LSTM**



•

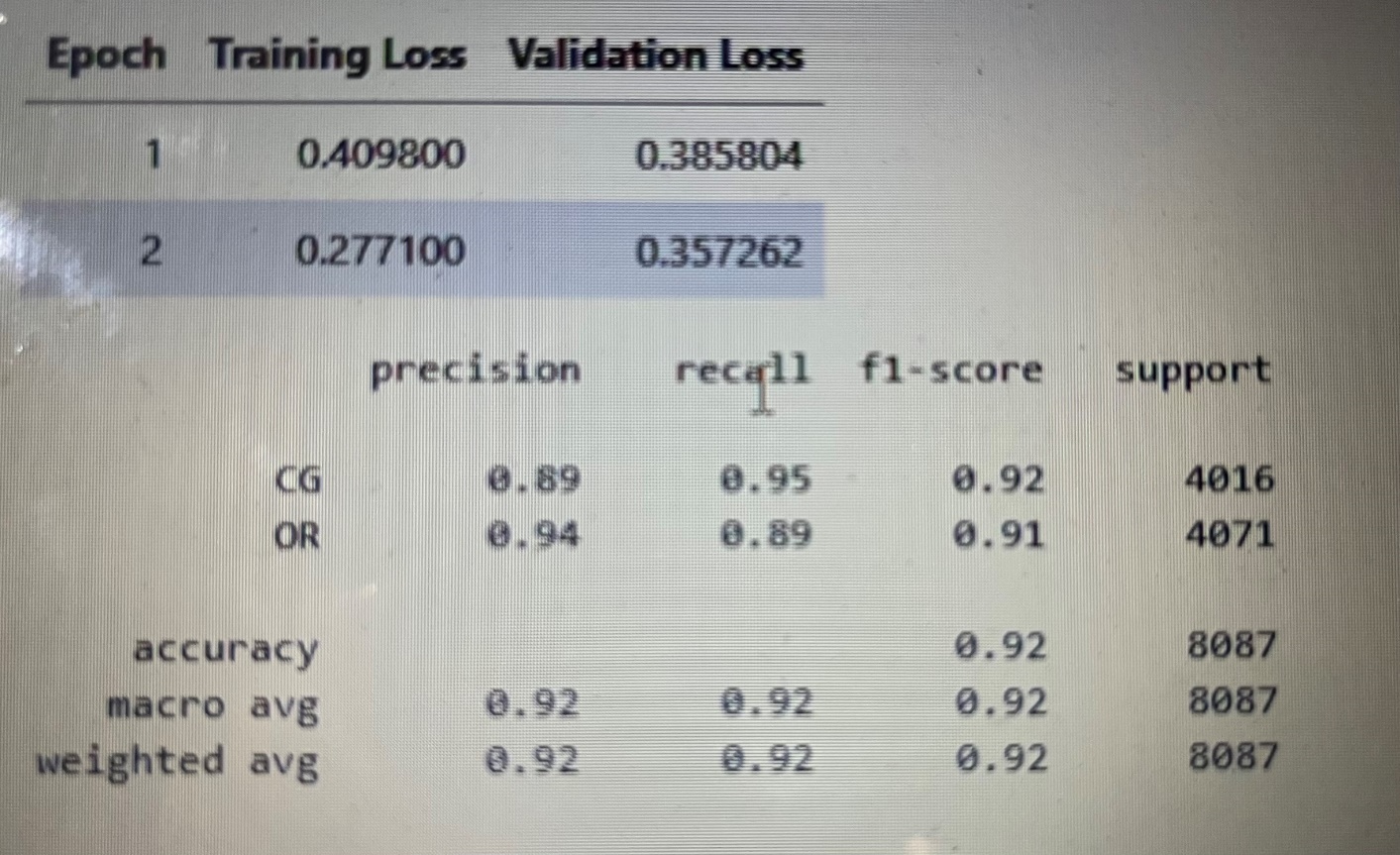
**• Performance of BILSTM**

****

***Conclusion:***

The BiLSTM model outperforms with the highest accuracy of 91% and balanced F1-scores ,compared to LSTM.

# BERT (Bidirectional Encoder Representations from Transformers)model for sequence classification using the Transformers library by Hugging Face



***Summary***

• The model achieved excellent performance, with an overall accuracy of 92%.

• Precision, Recall, and F1-Score for both classes (CG and OR) are all around 0.92, indicating balanced and strong performance across both classes.

• The training loss decreased rapidly, showing that the model is converging well, though the validation loss slightly increased in the last epoch, which could suggest mild overfitting.

• This model is performing well on the classification task and generalizes well to unseen data based on the validation results.