Report on Real vs. Fake Fact Classification for Chennai

Problem Setup: The goal of this study is to classify statements about Chennai as either real or fake. The dataset consists of 100 real facts and 100 fake facts generated using Chatgpt 4o model. The task is framed as a binary classification problem, with various preprocessing techniques and machine learning models applied to achieve optimal performance. The dataset was split into 80% training and 20% testing after labelling the dataset with 1 for real facts and 0 for fake facts. Later validation set and cross validation has been applied for Manual Hyperparameter tuning. Totally different sets of 7 experiments were conducted.

Preprocessing Techniques used: Lemmatization and Stemming (Reduced words to their root forms), Punctuation Removal, Stopword removal, Expand contractions, lowercasing, tokenization.

Feature Extraction and experiment setup: ***TF-IDF*** vectorization was applied (unigrams and bigrams) to capture word presence and contextual patterns. Models Selected were - ***Logistic Regression*** (Experiments 1, 4, 5, and 6), ***Support Vector Machine*** (SVM) (Experiments 2 and 2.1), ***Naive Bayes*** (MultinomialNB) (Experiments 3 and 3.1). *GridSearchCV* was used to tune hyperparameters with parameter grid 'C': [0.1, 1, 10] for Logistic regression and SVM and 'alpha': [0.1, 0.5, 1, 2], manual tuning was also performed with cross-validation.

Experimental Results, Interpretation and Key Takeaway:

1. **Lemmatization vs. Stemming**: In Exp 1, Both lemmatization and stemming yielded the same performance in terms of accuracy (**0.78**) with Logistic Regression in which **'c'** was set to **0.1** using GridSearchCV. For this dataset, the aggressive reduction of words to their stems did not significantly degrade model performance, suggesting that either approach is suitable. Lowecasing, tokenization and stopword removal was also performed in preprocessing.

2. **SVM Performance**: In exp 2, SVM performed similarly to Logistic Regression when stopwords were removed, achieving an accuracy of **0.78** and best 'c' was again found to be 0.1 using GridSearchCV. This shows that both models handle text classification well with lemmatization. Lowecasing, tokenization and stopword removal was also performed.

3. **Stopword Retention**: In Exp 2.1, where stopwords were retained, **SVM** achieved a much higher accuracy of **0.93 (C=1)**. This suggests that stopwords provide useful contextual information, helping the model improve its precision and recall for both real and fake facts. By retaining common words, the model can capture patterns that are crucial for understanding the structure of factual statements.

4. The Naive Bayes model performed similarly to Logistic Regression and SVM with an accuracy of **0.78 in Exp 3**. Although Naive Bayes assumes feature independence, which is not always realistic in text classification, it still performed well with TF-IDF features. The model slightly favored predicting fake facts (higher recall for class 0), but it struggled a bit with classifying real facts, as indicated by the lower recall for class 1.

5. Retaining stopwords and removing punctuation (Experiment 3.1) led to a substantial improvement in accuracy (**0.88** vs. **0.78**), best alpha was found to be 1 using tuning. This suggests that stopwords, while often considered noise, contribute important contextual information that helps the model differentiate between real and fake facts. Retaining stopwords, as in Experiment 3.1, allowed the NB model to capture dependencies between common words, leading to better predictions.

6. In exp 4, introducing additional preprocessing steps (such as **contraction expansion**, **punctuation removal**, and **keeping stopwords**) resulted in the highest test accuracy of **0.90**. The model was well-balanced between identifying both real and fake facts, with a slightly higher precision for fake facts and a higher recall for real facts. Expanding contractions contributed to better text

normalization, allowing the model to better understand word relationships, especially in short and factual sentences.

7. In **Exp 5**, using **SelectKBest** to reduce the number of features (from the full feature set to the top 500) improved the model's ability to focus on the most important words and n-grams in the dataset further improving the model's accuracy to **0.93.** This helped reduce noise and enhanced generalization. This experiment suggests that dimensionality reduction, when applied carefully, can enhance the performance of Logistic Regression by focusing on the most critical aspects of the data.

8. **Exp 6** used a fixed validation set, achieving a **0.825** validation accuracy and **0.90** test accuracy with C=1. **Exp 7** used **cross-validation**, yielding higher cross-validation accuracy (**0.88125**) with C=10 and matching the test accuracy of **0.90**. This suggests that cross-validation is a more reliable method for hyperparameter tuning because it evaluates the model on multiple splits of the training data, which reduces the risk of overfitting to a single validation set.

Limitations:

1. Small Dataset: The dataset of 200 facts limits the generalizability of the findings. A larger dataset might allow more complex patterns to emerge.
2. Feature Engineering: While TF-IDF was effective, exploring more advanced techniques like word embeddings might yield better performance.
3. Assumptions: The model assumes that fake facts can be identified based on linguistic features alone, which may not always hold true in real-world scenarios where facts often require external verification.
4. Generalizability: While the results apply well to this dataset, extending the findings to broader domains (e.g., detecting fake news or misinformation) may require more sophisticated models and contextual understanding. The assumption that text-based classification alone can separate real from fake facts may not generalize to all types of information.