COMP 550 Programming Assignment 1

1 Problem Setup

This project aims to distinguish between facts & fake statements regarding cities. The challenge lies in developing a model that can accurately classify statements as true or false based on textual features. The main challenge lies in preprocessing decisions, selection of classification models with their hyper-parameters and so on. This task is essential for combating misinformation, particularly in an era where disseminating false information can significantly impact public perception and decision-making.

2 Dataset Generation & Experimental Procedure

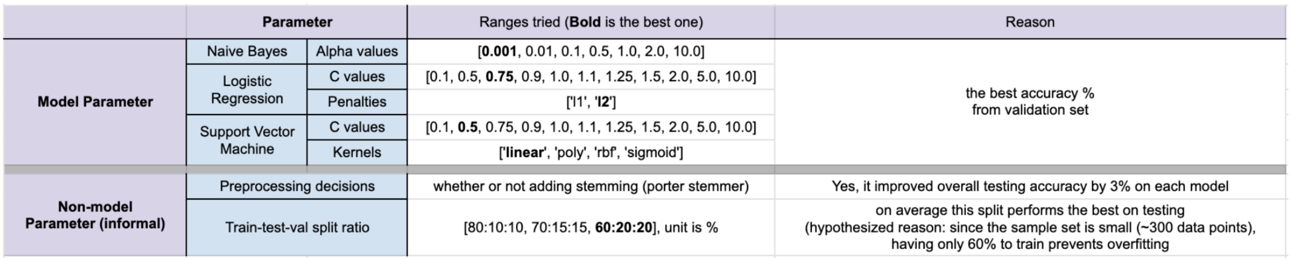
The dataset was generated from prompting ChatGPT for 170 facts and 150 fakes about Osaka, Japan. The following report is mainly based on the data of this single city. However for additional experiments to see what the effect is when adding more cities, I added 600 more facts/fakes from 3 students on Ed: Charles Blancas(Cebu City), David Wardan(Beirut), Bronwyn Walsh(Chicago). This finding is touched briefly in Section 4.

Preprocessing & feature extraction. The data was then preprocessed through several steps, including lowercasing, punctuation removal, stop word removal, lemmatization, and stemming. This aimed to normalize the text and reduce dimensionality for better feature extraction. Feature extraction was performed using the TF-IDF method since it also takes the importance of words into account as compared to CountVectorizer [2]. The dataset was subsequently split into training, validation, and test sets using various ratios, specifically 60% training, 20% validation, and 20% test.

Model hyperparameter selection, training & testing. The experiments were done using 3 models: Multinomial Naive Bayes (decision [1] inspired by Singh et. al's work[1]), Logistic Regression, and Support Vector Machine (SVM). Each model was trained and validated to find the optimal parameters based on validation accuracy, then reported performance on test accuracy. Here since we have a class-balanced dataset, rather than precision-recall, accuracy is a good measurement of performance.

3 Range of Parameter Settings Tried

There are 2 parts to this section: non-model & model parameters, where model parameters are the ones I followed the guideline of hyperparameter tuning using the validation set, while the non-model ones are informal & just serve as extra experiments to decide "which improves the test accuracy even more". Details are shown in Figure 1.



	Parameter		Ranges tried (**Bold** is the best one)	Reason
Model Parameter	Naive Bayes	Alpha values	[**0.001**, 0.01, 0.1, 0.5, 1.0, 2.0, 10.0]	the best accuracy % from validation set
	Logistic Regression	C values	[0.1, 0.5, **0.75**, 0.9, 1.0, 1.1, 1.25, 1.5, 2.0, 5.0, 10.0]	
		Penalties	['l1', '**l2**']	
	Support Vector Machine	C values	[0.1, **0.5**, 0.75, 0.9, 1.0, 1.1, 1.25, 1.5, 2.0, 5.0, 10.0]	
		Kernels	['**linear**', 'poly', 'rbf', 'sigmoid']	
Non-model Parameter (informal)	Preprocessing decisions		whether or not adding stemming (porter stemmer)	Yes, it improved overall testing accuracy by 3% on each model
	Train-test-val split ratio		[80:10:10, 70:15:15, **60:20:20**], unit is %	on average this split performs the best on testing (hypothesized reason: since the sample set is small (~300 data points), having only 60% to train prevents overfitting

Figure 1: Model & non-model parameters tried

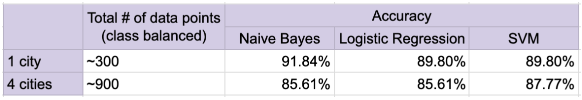
4 Results & Conclusions

Using the above validation set tuned-parameters, we achieved the following accuracy as shown in Figure 2. In addition, I also included the additional results when using the 4-city dataset as described in Section 2, following the same preprocessing techniques.

Naive Bayes is particularly effective for the simpler dataset (1 city), which may indicate that the assumptions of independence it makes are valid in that context. Logistic Regression seems to perform consistently across both datasets, making it a reliable choice when class balance is maintained. SVM might require further tuning or feature engineering in this project, as it performed the worst in both scenarios.

[1] chose Multinomial to make it more robust, also it can reduce to a Bernoulli if needed.

The accuracy of all models decreased when moving from the 1-city to the 4-city dataset. Naive Bayes, while still the best performer in both cases, showed a significant drop in accuracy, indicating that it may be sensitive to the increased complexity.



	Total # of data points (class balanced)	Accuracy		
		Naive Bayes	Logistic Regression	SVM
1 city	~300	91.84%	89.80%	89.80%
4 cities	~900	85.61%	85.61%	87.77%

Figure 2: Test accuracy comparison between single-city dataset & 4-city dataset.

5 Limitations

While the study provides promising results, several limitations must be considered. The dataset's size and diversity are crucial for generalizing the conclusions to a broader context of separating real vs. fake facts as seen in Section 4. The reliance on a limited set of facts and fakes may not capture the full spectrum of misinformation.

If generalizing this research question in real life, some assumptions we made include the idea that the words being used are relevant for distinguishing facts vs. facts are valid. However, it wouldn't be representable to generalize the conclusions of my experiments to the overall problem of separating real vs. fake facts for cities as 2 unrealistic assumptions are being made here: 1) There are only max 4 cities in "my world": almost all cities mentioned in the test set have been trained on, but in real situations, there can be rare cities that may only have a few data points so a model may be tested on without being trained on. To improve this, increasing the data size can be a solution but it's challenging. 2) Real data wouldn't necessarily be class-balanced (ex. depending on the platform scraped from, there could be many more facts than fakes); in this case, simple classifiers like Naive Bayes may be too biased. 3) The assumption that features are independent, which underlies Naive Bayes, may not hold in the real-world, particularly in complex datasets involving geographical and contextual information.

In conclusion, while the models demonstrate a reasonable ability to classify facts vs. fakes, further research with a more diverse dataset and additional features (such as text source credibility) is necessary to enhance robustness and applicability across different domains.

References

[1] Gurinder Singh, Bhawna Kumar, Loveleen Gaur, and Akriti Tyagi. Comparison between multinomial and bernoulli naïve bayes for text classification. pages 593–596, 04 2019.

[2] Sachin Soni. Naïve bayes machine learning algorithm: From basic to advanced, 2021. Accessed: 2024-09-27.