CSE 6363-001 -MACHINE LEARNING

Final Project – Bayesian Sentiment Predictor.

Name: Harini Aluka

ID:1002080841

Introduction:

The purpose of this project is to develop a Bayesian sentiment predictor using the wisesight dataset. Sentiment analysis is a process of identifying and classifying opinions expressed in a piece of text as positive, negative or neutral.

The wisesight dataset is publicly available dataset containing user reviews in Thai language from various domains. https://huggingface.co/datasets/wisesight_sentiment

Problem Addressed:

The problem is addressed by applying Bayesian model to multi class sentiment classification. Given dataset of text documents is divided into 3 sets as training set, test set, and validation set. Prior and conditional probabilities are used to classify a given text by assigning to one the three classes with highest probabilities.

Implementation:

The following steps were used to address the problem.

- Data cleaning: The dataset was cleaned to remove any irrelevant data such as punctuation marks, stop words and special characters.
- Data Pre-processing: The cleaned data is pre-processed using tokenization, stemming, and lemmatization techniques.
- Feature Extraction: The pre-processed data was converted into numerical features using various techniques such as bag-if-words, tf-idf , and word embeddings.
- Calculating prior probability: Calculating the prior probabilities of each class in the training set by counting the number of occurrences of each class and dividing by the total number of samples.
- Calculating Conditional probabilities: calculate the conditional probabilities of each word given each class.
- Classifying the validation and test sets: This function classifies a given 'text' into one of the classes based on the maximum score calculated using Naive Bayes algorithm. It takes in the pre-processed text, vocab (vocabulary of words present in the training set), prior_prob (prior probabilities of each class in the training set) and cond_prob(conditional probabilities of each word given its class in the training set). First, the function tokenizes the given 'text' using preprocess() function.

- Then, the function initializes a dictionary score with the prior probabilities of each class and iterates over each class and each token in the 'tokens' list. If the token is present in the 'vocab', it multiplies the current score of the class by the conditional probability of the token given its class. Finally, it returns the label of the class with the maximum score as the predicted class for the given text.
- Model Evaluation: The performance of each model was evaluated based on various metrics such as accuracy, precision, recall, and F1-score.

Results:

The trained models were evaluated on the wiseSight dataset, and the following results were obtained:

Accuracy: 0.5635018495684341

Precision: 0.7655236329935126

F1 Score: 0.8588510034668845

Recall: 0.9780935464192958

Interpretation of Results:

These results demonstrate the effectiveness of the Naive Bayes classifier in sentiment classification tasks.

The precision, recall, and F1-score values are high, indicating that the models are precise and have low false positives and false negatives. This project demonstrates that sentiment analysis can be performed on Thai language using the wiseSight dataset, which can be useful for businesses to analyse customer feedback and improve their products or services.

The accuracy 0.563 is low because of the in balanced class distribution. where one sentiment class may dominate the data, this can lead to biased predictions and lower accuracy.

When probabilities underflow, it can lead to numerical instability and loss of precision. In many programming languages and numerical libraries, the smallest representable positive floating-point number is on the order of 10^-308 or 10^-324, depending on the data type used. If a probability becomes smaller than this threshold, it will be rounded down to zero, leading to an inaccurate calculation. **Log is used in order to avoid such cases .**

If there is more than one sentiment class with the maximum score, indicating a tie, then it can randomly select one class from those classes.

Most of the texts in the test set are consistently predicting as class 1 this is due to the class imbalance. If the classes in the dataset are imbalanced, meaning some classes

have significantly fewer instances than others, the model may exhibit a bias towards the majority class. This can lead to lower accuracy.

Instructions to run the code:

Upload (finalproject.ipynb) file in google colab notebook.

Run the code.