5/1/2024

Project Report

CSCI-6366: Data Mining & Warehouse

Team Members:

Kavya Marthineni (20668298)
Om Parkash (20671728)
Suyesh Bhatta (20672078)

Data Preprocessing:

The methods we employed for data processing:

- **Tokenization:** The function ‘word_tokenize(text)’ provided by NLTK (Natural Language Toolkit) is used to split input data into individual words or tokens.
- **Lowercasing and Punctuation Removal:** List comprehensions are used to convert each token to lowercase and remove tokens that are not alphanumeric (Punctuation).
- **Stop word Removal**: The function ‘stopwords.words('english')’ from NLTK is used to retrieve stop words and thereby stop words are removed from the input data.
- **Stemming:** ‘PorterStemmer()’ from NLTK is used to perform stemming which reduces each token to its root or base form, which helps consolidate variations of words with the same meaning.
- **Feature Selection:** A correlation matrix is calculated which summaries the relation between multiple attributes. Having said that, the attributes like ‘coffee_id’, ‘roast’ and ‘roaster’ are dropped as they do not play a significant role in predicting the class of the coffee and retaining those attributes may increase data complexity.

Method and Implementation:

- Rows with missing values (NaN) are removed from the training data and categorical attributes like ‘origin’ are converted into numerical values using label encoding.
- ‘CountVectorizer’ is used to learn the vocabulary of the text data by fitting it on the training data's “review” column and processed reviews are then converted to Bag-of-Words features which is a sparse matrix where each row represents a review, and each column represents a word in the review. Cell values indicate word frequencies.
- This DataFrame is then merged with the original training data to create a new dataframe containing both original features and BoW features making it suitable for analysis and model training.

Evaluation and Cross-Validation:

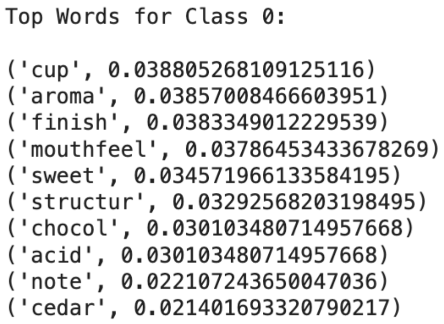
We employed K-Fold cross-validation with 5 folds, meaning the dataset was divided into 5 equal parts. Within each fold, we trained a Multinomial Naive Bayes classifier on the training subset and evaluated its performance on the validation subset using the F1 score metric. The F1 score is a measure of a model's accuracy that considers both the precision and recall of the model's predictions.

Here are the F1 scores for training and validation

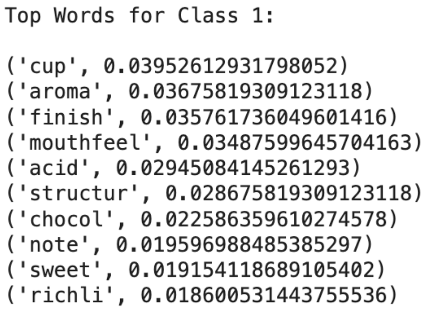
F1	**Training F1 Score**	**Validation F1 Score**
1	0.970	0.937
2	0.976	0.948
3	0.973	0.92
4	0.968	0.976
5	0.976	0.919

Insightful Findings:

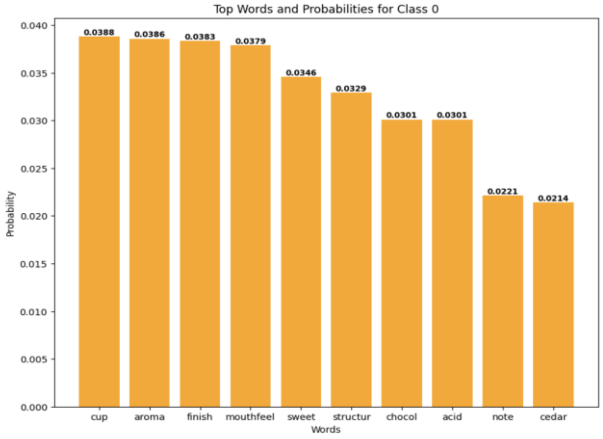
The following screenshot specifies the words from the provided reviews along with their corresponding probabilities which make significant contribution in distinguishing the Coffee as Class 1 and Class 0 respectively.

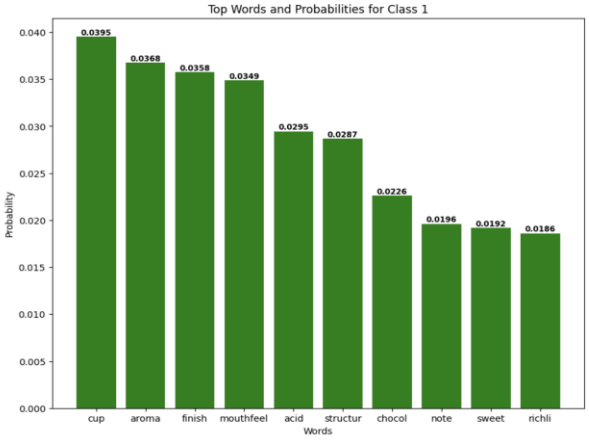


```
Top Words for Class 0:

('cup', 0.038805268109125116)
('aroma', 0.03857008466603951)
('finish', 0.0383349012229539)
('mouthfeel', 0.03786453433678269)
('sweet', 0.034571966133584195)
('structur', 0.03292568203198495)
('chocol', 0.030103480714957668)
('acid', 0.030103480714957668)
('note', 0.022107243650047036)
('cedar', 0.021401693320790217)
```



```
Top Words for Class 1:

('cup', 0.03952612931798052)
('aroma', 0.03675819309123118)
('finish', 0.035761736049601416)
('mouthfeel', 0.03487599645704163)
('acid', 0.02945084145261293)
('structur', 0.028675819309123118)
('chocol', 0.022586359610274578)
('note', 0.019596988485385297)
('sweet', 0.019154118689105402)
('richli', 0.018600531443755536)
```





Challenges:

- We faced the problem during preprocessing because there are multiple ways to do the text data preprocessing using TF-IDF or bag-of-words and more. We tried both but with TF-IDF we got less accuracy for training data then we tried bag-of-words preprocessing method which gave us the better accuracy.
- We also face the problem how to retrieve the best model from all k-folds to train on entire dataset the naïve bayes library that we used in our code we couldn't find how to retrieve best model from that but then we figure out that automatically that library choose the best model by default so we tried different method to over come this problem using for loop each k-fold dataset is trained and if the model is better than previous one then it will replace that model with better one that's how we saved best k-fold model and trained whole dataset on that model.
- Tried to calculate log probabilities of features while retrieving the highly dominant words from the reviews but couldn't do it efficiently. Then tried calculating the direct probabilities of features which served our objective.
- Tried multiple ways to improve the accuracy using normalization with min-max method and StandardScaler but the accuracy was same.

Group Contributions:

Om Parkash and Suyesh Bhatta: Preprocessed the data with multiple approaches (Bag-of Words, TF-IDF) and training the Naïve Bayes Model with different libraries.
Kavya Marthineni: Retrieving the highly dominant words from the reviews and making Report.

References:

1. Removing stop words with NLTK in Python
https://www.geeksforgeeks.org/removing-stop-words-nltk-python/#

2. Text Preprocessing in Python: Steps, Tools, and Examples
https://medium.com/product-ai/text-preprocessing-in-python-steps-tools-and-examples

3. Naive Bayes Classifier in Python (K-Fold Cross Validation)
https://www.kaggle.com/code/prashant111/naive-bayes-classifier-in-python

4. Naive Bayes on Amazon fine food review data
https://www.kaggle.com/code/sumansourav/naive-bayes-on-amazon-fine-food-review-data

5. How cross-validation can go wrong and how to fix it
https://towardsdatascience.com/how-cross-validation-can-go-wrong-and-how-to-fix-it

6. Get Started with Naive Bayes Algorithm: Theory & Implementation
https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm

7. How to select the best model using cross validation in python
https://www.youtube.com/watch?v=Bcw8S449QW4

8. K-Fold Cross Validation - Intro to Machine Learning
https://www.youtube.com/watch?

9. Calculating correlation matrix
https://medium.com/@shuv.sdr/na%C3%AFve-bayes-classification-in-python-f869c2e0dbf1