Report on text classification

Author - Pranjal Dhakal
Email – pranjal750@gmail.com
Tel - +977-9849958393

Problem statement:

In this assignment, the task is to build a Machine learning model in order to classify text samples in testing dataset into their corresponding classes. There are a total of 12 classes. For this task, I have used 2 models and trained them using a variety of features extracted from the training dataset. A comparison of performance among the models is made and presented in this document. Accuracy and F1-score are used for the quantitative measure of performance.

Training set and testing set creation

The training dataset consisting of 23615 samples is split in a (20:80) ratio to form the testing and training datasets. Stratified sampling is used to ensure that the distribution of samples across the 12 labels in the testing and training dataset is identical.

Features used

The text in the training dataset cannot be used to train the Machine learning models in its native form. So, feature extraction is performed from the training text. Feature extraction reduces the dimension of the training input and removes insignificant features thereby reducing the needless computations and increasing performance of the Machine learning models. These features are then fed into the models for training.

The features that are used for this task are:

Count Vectors

Count Vector is derived as one of the feature in this task. 13924 attributes are derived for each of the training and testing sample. This results in training feature dataset to be of dimension (18892, 13924) and that for testing is (4723, 13924).

TF-IDF (term frequency-inverse document frequency)

a. Word level
b. N-gram level
c. Character level

5000 attributes are derived using TF-IDF as feature for each of the training and testing sample. The training feature dataset dimension is (18892, 5000) and that for testing is (4723, 5000).

Models used

I have chosen some popular yet relatively simpler machine learning models to classify texts for this task. The models that I have used are:

1. Multinomial Naïve Bayes Model
2. Logistic Regression Model
3. Prediction using mode of all other models

    In this third method, the prediction using all the other models and features (8 combinations) was used. The mode (highest frequency) was derived from the 8 predictions for each test sample and final prediction was determined.

Performance measures

The parameters used to measure the performance of the Machine learning models for this task are:

Accuracy

Accuracy = (TP+TN) / (P+N)

F1 score

F1 score = 2*(precision* recall) / (precision+ recall)

Where, precision = TP / (TP+FP)

And, recall = TP / (TP+FN)

Result Analysis

The training process was run multiple times dividing the total training data into testing and testing datasets in the same (20-80) ratio. The split was made randomly every time using stratified sampling.

The performance measure for when random state was chosen 42 is shown below in tabulated form. Similar trend was observed upon changing the random state.

		Features			
Models	**F1 score (%)**	*Count Vectors*	*TF-IDF (word)*	*TF-IDF (n-gram)*	*TF-IDF (character)*
	NB Multinomial	89.06	90.09	82.15	86.46
	Logistic Regression	93.51	91.82	84.41	92.24
	Mode Prediction	92.18			

Table1: F1-score comparison for different features and models.

		Features			
Models	**Accuracy (%)**	*Count Vectors*	*TF-IDF (word)*	*TF-IDF (n-gram)*	*TF-IDF (character)*
	NB Multinomial	88.43	89.41	80.45	85.24
	Logistic Regression	93.01	91.21	82.46	91.74
	Mode Prediction	91.63			

Table2: Accuracy comparison for different features and models.

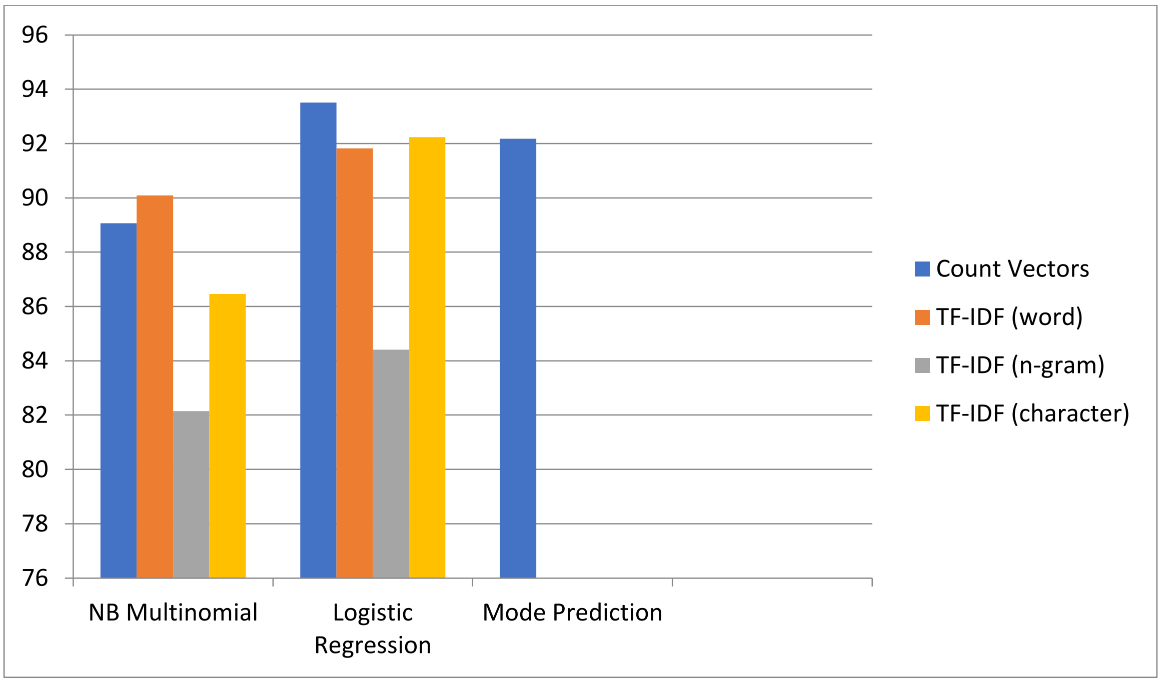

Fig1: F1-score comparison for different features and models.

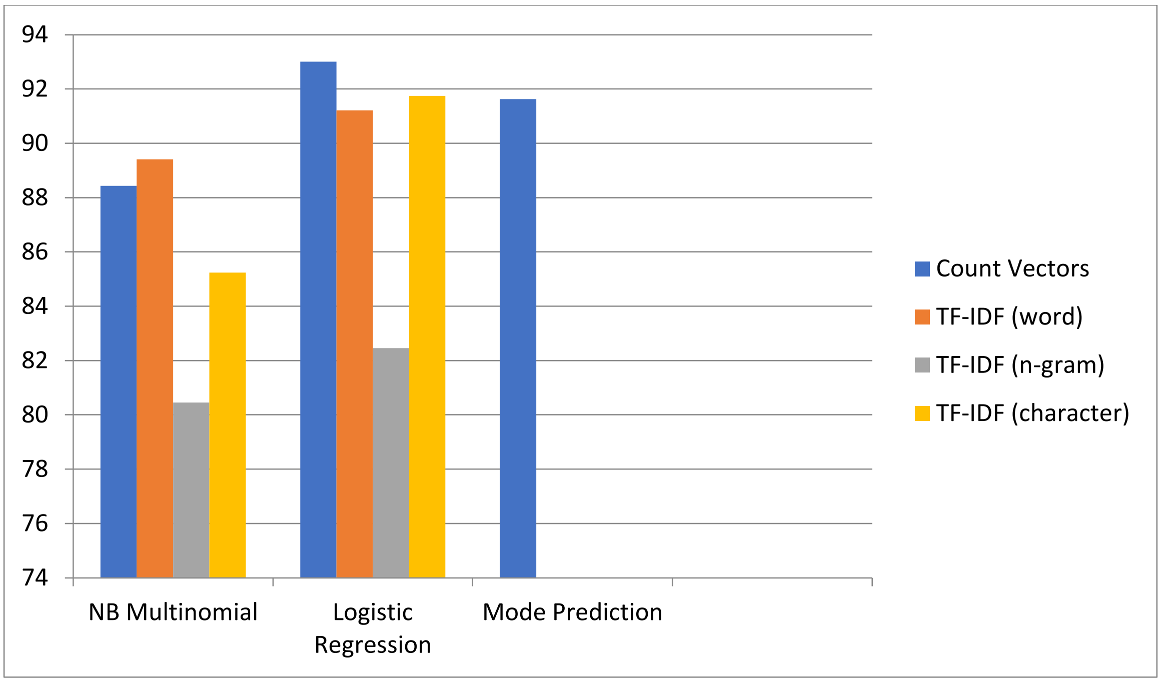

Fig2: Accuracy comparison for different features and models.

Conclusion

We can see that Logistic regression model using count vector as training feature repeatedly outperformed other models with other features.

So, I decided to use logistic regression model with count vector as training feature to determine the output labels of the given testing dataset.

References

- Bansal, S., & Natural Language Processing and Machine Learning. (2018, April 23). A Comprehensive Guide to Understand and Implement Text Classification in Python. Retrieved from https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/
- Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques*. Amsterdam: Morgan Kaufmann.