Assignment A02 ITAI 1371

Contributors: Jiri Musil, Martin Demel, Jasmine Campos, Munachimso Benson

Department of Science, Technology, Engineering & Math, Houston Community College

ITAI - 1371 Introduction to Machine Learning

Professor Vishwa Rao

September 16th, 2025.

**Introduction**

We explored 2 notebooks to develop a clear text‑classification workflow. Our goal was to convert raw text into numerical features and then train and evaluate a sentiment model on Amazon product reviews. The first notebook focused on understanding and verifying Bag of Words encodings on a small corpus so we could see exactly how vocabularies and vectors are created. The second notebook built a practical processing pipeline and assessed model performance on a held-out split. Going through both parts together helped us confirm each step and connect representation choices to the results.

**Body**

In the Bag of Words notebook, we represented 3 short sentences that include the first, second, and third documents using scikit-learn. We applied CountVectorizer in both binary and raw count modes, then switched to TfidfVectorizer configured without inverse document frequency to produce pure term frequency, and finally with IDF to compute TF-IDF. The notebook prints the learned vocabulary and the matrices, and it highlights two implementation details from scikit-learn.

In the KNN notebook, we loaded AMAZON‑REVIEW‑DATA‑CLASSIFICATION.csv with 70,000 rows and 6 columns, including reviewText, summary, verified, time, log\_votes, and isPositive. The label counts were 43,692 positive and 26,308 negative. We noticed missing values in reviewText (12) and summary (15). Text preparation used NLTK. We retained key negations, tokenized, removed other stop words and numeric or very short tokens, and applied the English SnowballStemmer. We split the data 90% for training and 10% for validation with random\_state=324. A scikit-learn pipeline was created by combining CountVectorizer(binary=True, max\_features=15) with KNeighborsClassifier. On the validation set, we observed the confusion matrix [[1325 1280], [902 3493]] and an accuracy of 0.688. The classification report showed a precision of 0.59 and a recall of 0.51 for the negative class, and a precision of 0.73 and a recall of 0.79 for the positive class.

**Conclusion**

By working together, we have linked the feature encodings from the first notebook to a complete baseline classifier in the second. With only fifteen binary features, the KNN model achieved about 69% accuracy, demonstrating how representation choices directly influence outcomes. The notebook also showed us concrete ideas for improvement within the same framework, such as trying different K values, switching to TF or TF-IDF, adjusting the vocabulary size, and adding simple textual cues like punctuation or capitalization.