Project Documentation

Overview/Function of the code

- Implemented in python with NLTK, pandas, numpy, matplotlib, sklearn, seaborn
- Four Jupyter/Google Colab notebooks (two notebooks with combination of models- VADER/HuggingFace and Naive Bayes/SVM):
    - VADER model with NLTK
    - HuggingFace

    - Naive Bayes Classifier
    - SVM with TF-IDF vectorization
- The code can be used for an introduction to the diverse machine learning models/implementation guide, and the tool can be used to test product reviews for various use cases as described below
- VADER model with NLTK
    - Download the data and input into pandas dataframe
    - Setup the Sentiment analyzer from NLTK
    - Calculate scores for all the reviews in the test set
    - Plot the results with matplotlib and seaborn
- NaiveBayes
    - Download the data and input into pandas dataframe
    - Extract review content, tokenize text data, and assign positive/negative labels based on ratings
    - After fitting, output key metrics: accuracy, precision, recall, and F1-score
- SVM:
    - Download the data and input into pandas dataframe
    - Extract review content, create TF-IDF features, and assign positive/negative labels based on ratings
    - After training, output key metrics: accuracy, precision, recall, and F1-score
- KAMAnalyzer:
    - Takes in written reviews and classifies them as positive, neutral, or negative using the techniques that we explored above.
    - Utilizes the best models that we explored(VADER and HuggingFace) to determine the sentiment
- Ideas for future extension:
    - Perform hyperparameter tuning to find the optimal settings for better accuracy and generalization
    - Incorporate cross-validation to get a more robust estimate of the model's performance
    - Conduct error analysis to understand common misclassifications and identify patterns the model struggles with

Use Cases

Sentiment analysis of any product review to determine positive or negative content that can be integrated into any of the following:

- E-Commerce Platforms: Automatically categorize positive and negative feedback to help buyers make informed decisions based on aggregated opinions
- Customer Support: Identify common issues or recurring themes in negative reviews to improve service quality
- Market Research: Review of competitors’ products to understand market trends
- Brand Monitoring: Monitor social media platforms and online forums for discussion about a brand, managing reputation

Software Implementation

Vader and Hugging Face Jupyter Notebook

- Data Exploration and Analysis:
    - Import necessary libraries such as pandas, numpy, matplotlib, seaborn, and nltk
    - Load amazon review dataset into a pandas DataFrame and perform exploratory data analysis
- NLP with NLTK:
    - NLTK (Natural Language Toolkit) is used for basic NLP tasks. The code tokenizes and performs part-of-speech tagging on a sample review from the dataset.
- Sentiment Analysis with VADER:
    - The VADER sentiment analysis tool from NLTK is employed to score the sentiment of each review in the dataset.
    - Calculated overall sentiment score and created a new DataFrame that combines sentiment scores with metadata.
- Visualization and Hugging Face Integration:
    - Visualizations of sentiment scores against user ratings using seaborn and matplotlib.
    - An example sentiment analysis pipeline from Hugging Face's Transformers library

Naive Bayes and SVM Google Colab Notebook

- Data Loading and Preprocessing:
    - Extracts the review content and converting the ratings into binary labels (positive or negative).
- Naive Bayes Sentiment Analysis:
    - Utilizes the CountVectorizer to convert the text data into a matrix of token counts.
    - Trains a Multinomial Naive Bayes classifier on the transformed data.
- TF-IDF Vectorization and SVM Classification:
    - Uses the TfidfVectorizer to create TF-IDF features from the review content.
    - Splits the TF-IDF vectors into training and testing sets.
    - Implements a Support Vector Machine (SVM) classifier with a linear kernel
- Result Display:

    - Prints the accuracy of both the Naive Bayes and SVM classifiers on their respective test sets.
    - Outputs key metrics such as precision, recall, and F1-score for positive and negative sentiments from the classification reports.

<u>KAMSentimentAnalyzer - Main Driver File</u>

- Utilizes the VADER and HuggingFace sentiment analyzers to determine whether or not a written review is positive, neutral or negative.
- Provides a user interface that asks for a review, and will output whether that review is positive, neutral or negative.

Installation and Setup/Software usage

Download repo from github

Install dependencies (recommended to use pip)

    - pip install scikit-learn, pip install seaborn, pip install matplotlib, pip install numpy, pip install nltk, pip install pandas

To run NaiveBayesAndSVM.ipynb:

- Import file to Google Colab
- Add amazon.csv to '/content/gdrive/MyDrive/data' in your Google Drive filepath
- Run each cell in order
- Enjoy!

To run VaderAndHuggingFace.ipynb:

- Import file to Jupyter Notebook
- Add amazon.csv to the data directory, which should be located in the same file path level of the ipynb file
- Run each cell in order
- Enjoy!

To run KAMSentimentAnalyzer.py:

- Make sure you have amazon.csv in a folder named data in the same directory as the py file
- Run the file directly using the following Unix command: python KAMSentimentAnalyzer.py
- When prompted, provide the review you would like analyzed. This could be from your own dataset using a similar format to the csv files provided in the repo or even a review from the Kaggle Amazon review dataset itself

Team Member Contribution:

Anish - Contributed to the development of Naive Bayes and SVM classifiers. This includes the preprocessing of the dataset and implementation of the classifiers themselves.
Matt - Contributed to the development of VADER model with nltk and the Hugging Face transformer model. This includes the exploration of the dataset using the nltk library and the exploration of the Hugging Face transformer.
Kevin - Aggregated results of all model development into KAM analyzer. This includes adapting models to work with a single imputed sentence and evaluating which models to use in the sentiment analyzer.