Semantic Classification Implementation Report

Objective

The goal is to classify business reviews as positive or negative using different machine learning models. My implementation covers data preprocessing, feature extraction, model training, and evaluation for best-possible accuracy.

Preprocessing

Taken Steps:

1. Remove Punctuation & Numbers: This will ensure that only meaningful text remains.
2. Convert to Lowercase: This will standardize the text.
3. Stopword Removal: The most common English stopwords were removed using NLTK to reduce noise.
4. Stemming: using PorterStemmer for normalizing words into their root form, such that "running" will become "run."
5. Reasoning: This is in order to reduce the dimensions and enhance model performance since it removes extraneous variation from the data.

Feature Extraction

TF-IDF Vectorization:

- Captures the importance of terms within a document relative to the corpus.
- Parameters:
  - `ngram_range=(1, 2)`: include unigrams and bigrams that capture contextual relationships.
  - `max_features=5000`: limits vocabulary size to balance performance and memory usage.

SMOTE (Synthetic Minority Oversampling Technique):

Class imbalance had been addressed by oversampling, also improving the recall on the minority class.

Models Trained

Four different machine learning models have been evaluated:

K-Nearest Neighbors (K-NN):

Underperformed because of the curse of dimensionality in high-dimensional TF-IDF data.

Decision Tree:

It achieved a moderate accuracy but was highly subject to overfitting.

Logistic Regression:

Parameters: Tuned hyperparameters including:

- C=10.0 (regularization strength).
- penalty='l2' (ridge regression).
- solver='liblinear' (suitable for smaller datasets).

Highest performance among other models, which is well-balanced between precision, recall, and F1-score.

Neural Network:

Performed well but required more computational resources.

Evaluation

Output:

```
Training K-NN...
Performance of K-NN on Validation Set:
              precision    recall  f1-score   support

           0       1.00      0.01      0.01      2114
           1       0.30      1.00      0.46       886

    accuracy                           0.30      3000
   macro avg       0.65      0.50      0.23      3000
weighted avg       0.79      0.30      0.14      3000

--------------------------------------------------
Training Decision Tree...
Performance of Decision Tree on Validation Set:
              precision    recall  f1-score   support

           0       0.85      0.84      0.84      2114
           1       0.62      0.64      0.63       886

    accuracy                           0.78      3000
   macro avg       0.73      0.74      0.74      3000
weighted avg       0.78      0.78      0.78      3000

--------------------------------------------------
Training Logistic Regression...
Performance of Logistic Regression on Validation Set:
              precision    recall  f1-score   support

           0       0.94      0.91      0.92      2114
           1       0.80      0.85      0.82       886

    accuracy                           0.89      3000
   macro avg       0.87      0.88      0.87      3000
weighted avg       0.89      0.89      0.89      3000

--------------------------------------------------
Training Neural Network...
Performance of Neural Network on Validation Set:
```

```
              precision    recall  f1-score   support

           0       0.93      0.92      0.92      2114
           1       0.80      0.82      0.81       886

    accuracy                           0.89      3000
   macro avg       0.87      0.87      0.87      3000
weighted avg       0.89      0.89      0.89      3000
```

Highest performance is the **Logistic Regression** model

Validation Metrics and Test Set Predictions

- Logistic Regression achieved best balance between precision and recall.
- It effectively handled imbalanced datasets with well-tuned hyperparameters and class-weight balancing.