Text Classification - NLP Report

Shubham Idekar
College of Engineering
Northeastern University
Boston, MA
idekar.s@northeastern.edu

Abstract

In this report, I explored the application of Multinomial Naive Bayes (NB) and Convolutional Neural Network (CNN) models for text classification in e-commerce, utilizing a robust dataset. Data preprocessing involved contraction expansion, punctuation removal, tokenization, lowercase conversion, elimination of numerical words, removal of stopwords, and stemming or lemmatization. The study aimed to categorize product descriptions into predefined classes. Multinomial NB served as a probabilistic baseline, while CNN, known for its capacity to capture intricate patterns in text, provided a more complex alternative.

1 Dataset

I used eCommerce dataset.

```
df
```

	Label	Text
0	Household	Paper Plane Design Framed Wall Hanging Motivat...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...
4	Household	Incredible Gifts India Wooden Happy Birthday U...
...	...	...
50420	Electronics	Strontium MicroSD Class 10 8GB Memory Card (Bl...
50421	Electronics	CrossBeats Wave Waterproof Bluetooth Wireless ...
50422	Electronics	Karbonn Titanium Wind W4 (White) Karbonn Titan...
50423	Electronics	Samsung Guru FM Plus (SM-B110E/D, Black) Colou...
50424	Electronics	Micromax Canvas Win W121 (White)

50425 rows × 2 columns

```
#Checking a sample value
df['Text'][0]
```

```
'Paper Plane Design Framed Wall Hanging Motivational Office Decor Art Prints (8.7 X 8.7 inch) - Set of 4 Painting made up in sy
nthetic frame with uv textured print which gives multi effects and attracts towards it. This is an special series of paintings
which makes your wall very beautiful and gives a royal touch. This painting is ready to hang, you would be proud to possess thi
s unique painting that is a niche apart. We use only the most modern and efficient printing technology on our prints, with only
the and inks and precision epson, roland and hp printers. This innovative hd printing technique results in durable and spectacu
lar looking prints of the highest that last a lifetime. We print solely with top-notch 100% inks, to achieve brilliant and true
colours. Due to their high level of uv resistance, our prints retain their beautiful colours for many years. Add colour and sty
le to your living space with this digitally printed painting. Some are for pleasure and some for eternal bliss.so bring home th
is elegant print that is lushed with rich colors that makes it nothing but sheer elegance to be to your friends and family.it w
ould be treasured forever by whoever your lucky recipient is. Liven up your place with these intriguing paintings that are high
definition hd graphic digital prints for home, office or any room.'
```

2 Preprocessing Data

2.1 Expanding Contraction

Contracted words are a common feature of natural language, especially in informal settings such as social media or messaging platforms.Contractions are shortened versions of words or phrases that are formed by combining two words and replacing one or more letters with an apostrophe. Examples of contractions include:

"can't" (from "cannot")
"won't" (from "will not")

```
df['Contractions'] = df['Text'].apply(lambda x: [contractions.fix(word) for word in x.split()])
df['No_contractions'] = [' '.join(map(str, l)) for l in df['Contractions']]
df.drop('Contractions',axis=1,inplace=True)
df.head()
```

	Label	Text	No_contractions
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...

Fig2.1: expanding contractions

2.2 *Remove punctuations*

Punctuation is often removed to simplify the analysis, and reduce the vocabulary size while preserving the meaningful content of the text.

We will use the punctuation library from the String package.

```
punc = string.punctuation
df['No_punc'] = df['No_contractions'].apply(lambda x: re.sub('[%s]' % re.escape(string.punctuation), '' , x))
df.head()
```

	Label	Text	No_contractions	No_punc
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF Floral Framed Painting Wood 30 inch x 10 i...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF UV Textured Modern Art Print Framed Painti...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting Synthetic 135...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...

Fig2.2 remove punctuations.

2.3 *Tokenization*

Tokenization is the process of breaking down text into individual words, phrases, or other meaningful elements, called tokens. We will use NLTK.word_tokenize() function to create a new column named "tokenized".

```
nltk.download('punkt')
df['Tokenized'] = df['No_punc'].apply(word_tokenize)
df.head()
```

```
[nltk_data] Downloading package punkt to C:\Users\Shubham
[nltk_data]     Idekar\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

	Label	Text	No_contractions	No_punc	Tokenized
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	[Paper, Plane, Design, Framed, Wall, Hanging, ...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF Floral Framed Painting Wood 30 inch x 10 i...	[SAF, Floral, Framed, Painting, Wood, 30, inch...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF UV Textured Modern Art Print Framed Painti...	[SAF, UV, Textured, Modern, Art, Print, Framed...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting Synthetic 135...	[SAF, Flower, Print, Framed, Painting, Synthet...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	[Incredible, Gifts, India, Wooden, Happy, Birt...

Fig2.3 Tokenization

2.4 *Convert to Lower Case*

All the alphabetic characters in a text are transformed to their corresponding lower case representation to reduce the vocabulary size and avoid duplication of words during text analysis.

```
df['Lower'] = df['Tokenized'].apply(lambda x: [text.lower() for text in x])
df.head()
```

	Label	Text	No_contractions	No_punc	Tokenized	Lower
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	[Paper, Plane, Design, Framed, Wall, Hanging, ...	[paper, plane, design, framed, wall, hanging, ...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF Floral Framed Painting Wood 30 inch x 10 i...	[SAF, Floral, Framed, Painting, Wood, 30, inch...	[saf, floral, framed, painting, wood, 30, inch...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF UV Textured Modern Art Print Framed Painti...	[SAF, UV, Textured, Modern, Art, Print, Framed...	[saf, uv, textured, modern, art, print, framed...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting Synthetic 135...	[SAF, Flower, Print, Framed, Painting, Synthet...	[saf, flower, print, framed, painting, synthet...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	[Incredible, Gifts, India, Wooden, Happy, Birt...	[incredible, gifts, india, wooden, happy, birt...

2.5 *Remove words containing digits*

Eliminating words that contain numeric characters from text analysis to reduce noise and improve the accuracy of language models. We will eliminate these words using Regular Expression.

```
df['No_num'] = df['Lower'].apply(lambda x: [re.sub(r'\w*\d\w*','',text) for text in x])
df.head()
```

	Label	Text	No_contractions	No_punc	Tokenized	Lower	No_num
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	[Paper, Plane, Design, Framed, Wall, Hanging, ...	[paper, plane, design, framed, wall, hanging, ...	[paper, plane, design, framed, wall, hanging, ...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF Floral Framed Painting Wood 30 inch x 10 i...	[SAF, Floral, Framed, Painting, Wood, 30, inch...	[saf, floral, framed, painting, wood, 30, inch...	[saf, floral, framed, painting, wood, , inch, ...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF UV Textured Modern Art Print Framed Painti...	[SAF, UV, Textured, Modern, Art, Print, Framed...	[saf, uv, textured, modern, art, print, framed...	[saf, uv, textured, modern, art, print, framed...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting Synthetic 135...	[SAF, Flower, Print, Framed, Painting, Synthet...	[saf, flower, print, framed, painting, synthet...	[saf, flower, print, framed, painting, synthet...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	[Incredible, Gifts, India, Wooden, Happy, Birt...	[incredible, gifts, india, wooden, happy, birt...	[incredible, gifts, india, wooden, happy, birt...

2.6 *Remove stopwords*

Process of eliminating common words such as "the", "a", "an", and "in" from text to reduce the dimensionality of the data, and to focus on the more meaningful words that carry the essence of the text.

We will use the stopwords library from the nltk module.

```
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
df['Stopwords_removed'] = df['No_num'].apply(lambda x: [word for word in x if word not in stop_words])
df.head()
```

```
[nltk_data] Downloading package stopwords to C:\Users\Shubham
[nltk_data]     Idekar\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

	Label	Text	No_contractions	No_punc	Tokenized	Lower	No_num	Stopwords_removed
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	Paper Plane Design Framed Wall Hanging Motivat...	[Paper, Plane, Design, Framed, Wall, Hanging, ...	[paper, plane, design, framed, wall, hanging, ...	[paper, plane, design, framed, wall, hanging, ...	[paper, plane, design, framed, wall, hanging, ...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	SAF Floral Framed Painting Wood 30 inch x 10 i...	[SAF, Floral, Framed, Painting, Wood, 30, inch...	[saf, floral, framed, painting, wood, 30, inch...	[saf, floral, framed, painting, wood, , inch, ...	[saf, floral, framed, painting, wood, , inch, ...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF 'UV Textured Modern Art Print Framed' Pain...	SAF UV Textured Modern Art Print Framed Painti...	[SAF, UV, Textured, Modern, Art, Print, Framed...	[saf, uv, textured, modern, art, print, framed...	[saf, uv, textured, modern, art, print, framed...	[saf, uv, textured, modern, art, print, framed...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting (Synthetic, 1...	SAF Flower Print Framed Painting Synthetic 135...	[SAF, Flower, Print, Framed, Painting, Synthet...	[saf, flower, print, framed, painting, synthet...	[saf, flower, print, framed, painting, synthet...	[saf, flower, print, framed, painting, synthet...
4	Household	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	Incredible Gifts India Wooden Happy Birthday U...	[Incredible, Gifts, India, Wooden, Happy, Birt...	[incredible, gifts, india, wooden, happy, birt...	[incredible, gifts, india, wooden, happy, birt...	[incredible, gifts, india, wooden, happy, birt...

Fig 2.6 Removed stopwords

2.7 *Stemming or Lemmatization*

Stemming and lemmatization are two techniques used in NLP to normalize words by reducing them to their base or root form; stemming chops off the end of words, while lemmatization uses a vocabulary and morphological analysis to reduce words to their canonical form.

Stemming: The stem of "running" is "run". Using a stemming algorithm, "running", "runs", and "runner" would all be reduced to the stem "run".

Lemmatization: The lemma of "running" is "run". Using a lemmatization algorithm, "running" and "runs" would be reduced to "run", while "runner" would be reduced to "run" as well, but only if the context suggests that it is being used as a verb. We will apply parts of speech tags, in other words, determine the part of speech (ie. noun, verb, adverb, etc.) for each word.

There are various stemmers and one lemmatizer in NLTK, the most common being:

Porter Stemmer from Porter (1980)

Wordnet Lemmatizer (port of the Morphy: https://wordnet.princeton.edu/man/morphy.7WN.html)

Action : We will apply NLTK’s Porter Stemmer within our trusty list comprehension.

```
: stemmer = PorterStemmer()
  df['Stem'] = df['Stopwords_removed'].apply(lambda x: [stemmer.stem(word) for word in x])
```

```
: # Displaying the original text and processed data
  df[['Text','Stem']]
```

	Text	Stem
0	Paper Plane Design Framed Wall Hanging Motivat...	[paper, plane, design, frame, wall, hang, moti...
1	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	[saf, floral, frame, paint, wood, , inch, x, ,...
2	SAF 'UV Textured Modern Art Print Framed' Pain...	[saf, uv, textur, modern, art, print, frame, p...
3	SAF Flower Print Framed Painting (Synthetic, 1...	[saf, flower, print, frame, paint, synthet, , ...
4	Incredible Gifts India Wooden Happy Birthday U...	[incred, gift, india, wooden, happi, birthday,...
...	...	...
50420	Strontium MicroSD Class 10 8GB Memory Card (Bl...	[strontium, microsd, class, , , memori, card, ...
50421	CrossBeats Wave Waterproof Bluetooth Wireless ...	[crossbeat, wave, waterproof, bluetooth, wirel...
50422	Karbonn Titanium Wind W4 (White) Karbonn Titan...	[karbonn, titanium, wind, , white, karbonn, ti...
50423	Samsung Guru FM Plus (SM-B110E/D, Black) Colou...	[samsung, guru, fm, plu, , black, colourblack,...
50424	Micromax Canvas Win W121 (White)	[micromax, canva, win, , white]

50424 rows × 2 columns

Fig 2.7 Stemming or lemmatization

3 Baseline Model- Multinomial Naïve Beyes

Here's a brief overview of how Multinomial Naive Bayes works for text classification:

Bag-of-Words Representation:

The first step is to represent the text data as a "bag of words," disregarding the order of words but considering their frequency. Each document is represented as a vector of word counts, where the elements of the vector correspond to the frequency of each word in the document.

Vocabulary Building:

The algorithm builds a vocabulary from the entire dataset, containing all unique words present in the corpus.

Probability Estimation:

MNB estimates the probability of each word occurring in a document for each class. It calculates the likelihood of observing each word given the class.

Prior Probability:

MNB also calculates the prior probability of each class, representing the likelihood of a document belonging to a particular class without considering the words.

Naive Bayes Assumption:

The "naive" assumption in Naive Bayes is that the features (words) are conditionally independent given the class label. While this assumption simplifies the model, it may not always hold true in practice.

Classification Decision:

Using Bayes' theorem, the algorithm calculates the posterior probability of each class given the document. The class with the highest posterior probability is assigned as the predicted class for the document.

Baseline Model - Naive Bayes

- Text Vectorization
  In order to perform machine learning on text data, we must transform the documents into vector representations. In natural language processing, **text vectorization** is the process ofconverting words, sentences, or even larger units of text data to numerical vectors.

```
In [21]: from sklearn.model_selection import train_test_split

         # Split the dataset into a training set and a testing set while preserving the class distribution
         X = df['Stem']  # Features
         y = df['Label']  # Target variable

         # Use stratified sampling to ensure the same class distribution in both sets
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

```
In [22]: from sklearn.feature_extraction.text import TfidfVectorizer

         # Convert the lists of words into strings
         X_train = X_train.apply(lambda word_list: ' '.join(word_list))
         X_test = X_test.apply(lambda word_list: ' '.join(word_list))

         tfidf_vectorizer = TfidfVectorizer()
         X_train = tfidf_vectorizer.fit_transform(X_train)
         X_test = tfidf_vectorizer.transform(X_test)
```

```
In [23]: from sklearn.naive_bayes import MultinomialNB

         nb_classifier = MultinomialNB()
         nb_classifier.fit(X_train, y_train)

Out[23]: ▸ MultinomialNB
         MultinomialNB()
```

Fig3.1: Implementing of Multinomial NB

Accuracy and ***Prediction*** of this model :

```
In [25]: from sklearn.metrics import classification_report, accuracy_score

         accuracy = accuracy_score(y_test, predictions)
         report = classification_report(y_test, predictions)
         print(f"Accuracy: {accuracy}")
         print(report)

Accuracy: 0.9411998016856717
                       precision    recall  f1-score   support

                Books       0.98      0.92      0.95      2364
Clothing & Accessories       0.98      0.94      0.96      1734
          Electronics       0.96      0.90      0.93      2124
            Household       0.90      0.98      0.94      3863

             accuracy                           0.94     10085
            macro avg       0.95      0.93      0.94     10085
         weighted avg       0.94      0.94      0.94     10085
```

Fig3.2: Model Performance by metrics.

Multinomial Naive Bayes Model Evaluation:
Precision:
Books: 0.98 - Among the instances predicted as "Books," 98% are correctly classified.
Clothing & Accessories: 0.98 - 98% of instances predicted as "Clothing & Accessories" are correct.
Electronics: 0.96 - 96% of instances predicted as "Electronics" are correct.
Household: 0.90 - 90% of instances predicted as "Household" are correct.

Recall:
Books: 0.92 - The model correctly identifies 92% of the actual instances of "Books."
Clothing & Accessories: 0.94 - 94% of instances of "Clothing & Accessories" are correctly identified.
Electronics: 0.90 - The model captures 90% of instances of "Electronics."
Household: 0.98 - An impressive 98% of instances of "Household" are correctly identified.

F1-Score:
Books: 0.95 - The harmonic mean of precision and recall for "Books."
Clothing & Accessories: 0.96 - The F1-score for "Clothing & Accessories."
Electronics: 0.93 - The harmonic mean of precision and recall for "Electronics."
Household: 0.94 - The F1-score for "Household."

Support:
Books: 2364 - There are 2364 instances of "Books" in the test set.
Clothing & Accessories: 1734 - There are 1734 instances of "Clothing & Accessories."
Electronics: 2124 - There are 2124 instances of "Electronics."

Household: 3863 - There are 3863 instances of "Household."

Accuracy:
The overall accuracy of the MultinomialNB model is approximately 94%, meaning it correctly classifies instances around 94% of the time.

Macro Avg and Weighted Avg:
Macro Avg: 0.94 - The average precision, recall, and F1-score across all classes without considering class imbalance.
Weighted Avg: 0.94 - Similar to macro avg, but considering the number of samples for each class, giving more weight to classes with more instances.

Interpretation:
The MultinomialNB model shows good performance across all classes. It performs particularly well in correctly identifying instances of "Books," "Clothing & Accessories," and "Household." The weighted average considers the class imbalance, providing a balanced overview of model performance.

4 Advanced model - CNN

Convolutional Neural Networks (CNNs) are primarily known for their success in image-related tasks, but they can also be adapted for text classification. Here's a simplified explanation of how a CNN model for text classification works:

Input Representation:
Text data is initially represented as numerical vectors, typically using techniques like word embeddings (e.g., Word2Vec, GloVe). Each word is represented by a vector, and a sequence of these vectors is used as the input.

Convolutional Layers:
Convolutional operations, which are highly effective in capturing local patterns, are applied to the input sequence. This involves using filters or kernels of fixed size that slide over the input.
The convolutional layer detects specific patterns or features in the input. In text, these features can represent n-grams or local word patterns.

Activation and Pooling:
After the convolution, an activation function (e.g., ReLU) is applied element-wise to introduce non-linearity.
Pooling layers (often max pooling) follow, reducing the dimensionality of the features while retaining the most salient information. This helps the model focus on the most important features.

Flattening and Fully Connected Layers:
The output from the convolutional and pooling layers is flattened into a one-dimensional vector.
Fully connected layers are then used to combine the learned features and make predictions. These layers can capture global relationships in the data.

Output Layer:
The final layer consists of one or more neurons representing the classes in the classification task. A softmax activation function is often used to convert the network's raw output into class probabilities.
Training:
The model is trained using labeled data with a specified loss function (e.g., categorical cross-entropy). The weights of the network are updated through backpropagation and optimization algorithms (e.g., Adam, SGD).

Prediction:

Once trained, the model can predict the class of a new text by passing it through the trained network, and the class with the highest probability is considered the predicted class.

```
In [24]: # Split the data into training and testing sets
train_data, test_data = train_test_split(df, test_size=0.2, random_state=42, stratify=df['Label'])

# Advanced Model: Convolutional Neural Network (CNN)
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(train_data['Label'])
y_test_encoded = label_encoder.transform(test_data['Label'])

tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_data['Stem'])
X_train_sequences = tokenizer.texts_to_sequences(train_data['Stem'])
X_test_sequences = tokenizer.texts_to_sequences(test_data['Stem'])

max_sequence_length = max(max(len(seq) for seq in X_train_sequences), max(len(seq) for seq in X_test_sequences))
X_train_padded = pad_sequences(X_train_sequences, maxlen=max_sequence_length, padding='post')
X_test_padded = pad_sequences(X_test_sequences, maxlen=max_sequence_length, padding='post')

embedding_dim = 50
vocab_size = len(tokenizer.word_index) + 1

cnn_model = Sequential()
cnn_model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=max_sequence_length))
cnn_model.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
cnn_model.add(GlobalMaxPooling1D())
cnn_model.add(Dense(64, activation='relu'))
cnn_model.add(Dropout(0.5))
cnn_model.add(Dense(len(label_encoder.classes_), activation='softmax'))

cnn_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
cnn_model.fit(X_train_padded, y_train_encoded, epochs=5, batch_size=64, validation_split=0.2)

cnn_accuracy = cnn_model.evaluate(X_test_padded, y_test_encoded, verbose=0)[1]
print("\nAdvanced Model (Convolutional Neural Network) Results:")
print(f"Accuracy: {cnn_accuracy}")
```

```
Epoch 1/5
505/505 [==============================] - 407s 802ms/step - loss: 0.4292 - accuracy: 0.8483 - val_loss: 0.1321 - val_accuracy:
0.9646
Epoch 2/5
505/505 [==============================] - 419s 829ms/step - loss: 0.0933 - accuracy: 0.9787 - val_loss: 0.0985 - val_accuracy:
0.9746
Epoch 3/5
505/505 [==============================] - 416s 824ms/step - loss: 0.0387 - accuracy: 0.9907 - val_loss: 0.1047 - val_accuracy:
0.9747
Epoch 4/5
505/505 [==============================] - 419s 829ms/step - loss: 0.0207 - accuracy: 0.9949 - val_loss: 0.1215 - val_accuracy:
0.9763
Epoch 5/5
505/505 [==============================] - 423s 837ms/step - loss: 0.0125 - accuracy: 0.9969 - val_loss: 0.1342 - val_accuracy:
0.9761

Advanced Model (Convolutional Neural Network) Results:
Accuracy: 0.9734258651733398
```

Evaluation metrics for this model:

```
316/316 [==============================] - 17s 54ms/step
Classification Report for CNN Model:
                        precision    recall  f1-score   support

                 Books       0.97      0.97      0.97      2364
Clothing & Accessories       0.98      0.98      0.98      1734
           Electronics       0.97      0.97      0.97      2124
             Household       0.97      0.97      0.97      3863

              accuracy                           0.97     10085
             macro avg       0.97      0.97      0.97     10085
          weighted avg       0.97      0.97      0.97     10085
```

CNN Model Evaluation:

Precision:

Precision is the ratio of correctly predicted positive observations to the total predicted positives. For each class, precision is calculated as TP / (TP + FP), where TP is the number of true positives and FP is the number of false positives. In your report, precision values range from 0.97 to 0.98, indicating high precision for all classes.

Recall:

Recall (or sensitivity or true positive rate) is the ratio of correctly predicted positive observations to the all observations in the actual class. For each class, recall is calculated as TP / (TP + FN), where TP is the number of true positives and FN is the number of false negatives. In your report, recall values range from 0.97 to 0.98, indicating high recall for all classes.

F1-Score:

F1-score is the weighted average of precision and recall. It considers both false positives and false negatives. For each class, F1-score is calculated as 2 * (precision * recall) / (precision + recall). In your report, F1-score values range from 0.97 to 0.98, indicating a good balance between precision and recall for all classes.

Support:

Support is the number of actual occurrences of the class in the specified dataset. It gives you an idea of how many samples in your test set belong to each class.

Overall Accuracy:

The accuracy is the overall correctly predicted instances divided by the total instances. In your case, the overall accuracy is around 97%, indicating the proportion of correctly classified instances.

Macro Avg and Weighted Avg:

Macro avg is the average of precision, recall, and F1-score across all classes without considering class imbalance. Weighted avg is the same as macro avg, but it considers the number of samples for each class, giving more weight to classes with more instances.

Interpretation:

The high precision, recall, and F1-score values for each class and the overall accuracy indicate that our CNN model is performing well on the test set.

References

[1] https://www.analyticsvidhya.com/blog/2020/12/understanding-text-classification-in-nlp-with-movie-review-example-example/

[2] https://www.kaggle.com/code/firozchowdury/data-pre-processing-e-commerce-dataset

[3] https://www.kaggle.com/code/bekkarmerwan/ecommerce-text-classification