```python
In [2]: import pandas as pd
        import numpy as np
        import re
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from sklearn.feature_extraction.text import TfidfVectorizer
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```python
In [3]: data = pd.read_csv(r'C:\Users\DELL\Downloads\train_product_data.csv')
```

```python
In [4]: # Display the first few rows of the dataset
        print(data.head())

        # Check for missing values
        print(data.isnull().sum())
```

```
                          uniq_id         crawl_timestamp  \
0  c2d766ca982eca8304150849735ffef9  2016-03-25 22:59:23 +0000
1  f449ec65dcbc041b6ae5e6a32717d01b  2016-03-25 22:59:23 +0000
2  0973b37acd0c664e3de26e97e5571454  2016-03-25 22:59:23 +0000
3  ce5a6818f7707e2cb61fdcdbba61f5ad  2016-03-25 22:59:23 +0000
4  29c8d290caa451f97b1c32df64477a2c  2016-03-25 22:59:23 +0000

                                         product_url  \
0  http://www.flipkart.com/alisha-solid-women-s-c... (http://www.flipkart.
com/alisha-solid-women-s-c...)
1  http://www.flipkart.com/aw-bellies/p/itmeh4grg... (http://www.flipkart.
com/aw-bellies/p/itmeh4grg...)
2  http://www.flipkart.com/alisha-solid-women-s-c... (http://www.flipkart.
com/alisha-solid-women-s-c...)
3  http://www.flipkart.com/alisha-solid-women-s-c... (http://www.flipkart.
com/alisha-solid-women-s-c...)
4  http://www.flipkart.com/dilli-bazaaar-bellies-... (http://www.flipkart.
com/dilli-bazaaar-bellies-...)

                                 product_name product_category_tree
\
0             Alisha Solid Women's Cycling Shorts           Clothing
1                                   AW Bellies           Footwear
2             Alisha Solid Women's Cycling Shorts           Clothing
3             Alisha Solid Women's Cycling Shorts           Clothing
4  dilli bazaaar Bellies, Corporate Casuals, Casuals           Footwear

               pid  retail_price  discounted_price  \
0  SRTEH2FF9KEDEFGF         999.0             379.0
1  SHOEH4GRSUBJGZXE         999.0             499.0
2  SRTEH2F6HUZMQ6SJ         699.0             267.0
3  SRTEH2FVVKRBAXHB        1199.0             479.0
4  SHOEH3DZBFR88SCK         699.0             349.0

                                          image  is_FK_Advantage_prod
uct  \
0  ["http://img5a.flixcart.com/image/short/u/4/a/...                 Fa
lse
1  ["http://img5a.flixcart.com/image/shoe/7/z/z/r...                 Fa
lse
2  ["http://img5a.flixcart.com/image/short/6/2/h/...                 Fa
lse
3  ["http://img6a.flixcart.com/image/short/p/j/z/...                 Fa
lse
4  ["http://img6a.flixcart.com/image/shoe/b/p/n/p...                 Fa
lse

                                         description     product_rating
\
0  Key Features of Alisha Solid Women's Cycling S...  No rating available
1  Key Features of AW Bellies Sandals Wedges Heel...  No rating available
2  Key Features of Alisha Solid Women's Cycling S...  No rating available
3  Key Features of Alisha Solid Women's Cycling S...  No rating available
4  Key Features of dilli bazaaar Bellies, Corpora...  No rating available

        overall_rating          brand  \
0  No rating available          Alisha
1  No rating available              AW
2  No rating available          Alisha
3  No rating available          Alisha
4  No rating available   dilli bazaaar
```

```
                                       product_specifications
0   {"product_specification"=>[{"key"=>"Number of ...
1   {"product_specification"=>[{"key"=>"Ideal For"...
2   {"product_specification"=>[{"key"=>"Number of ...
3   {"product_specification"=>[{"key"=>"Number of ...
4   {"product_specification"=>[{"key"=>"Occasion",...
uniq_id                           0
crawl_timestamp                   0
product_url                       0
product_name                      0
product_category_tree             0
pid                               0
retail_price                     57
discounted_price                 57
image                             3
is_FK_Advantage_product           0
description                       1
product_rating                    0
overall_rating                    0
brand                          4710
product_specifications            6
dtype: int64
```

In [4]:

```python
# Display basic information about the dataset
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   uniq_id                  14999 non-null  object
 1   crawl_timestamp          14999 non-null  object
 2   product_url              14999 non-null  object
 3   product_name             14999 non-null  object
 4   product_category_tree    14999 non-null  object
 5   pid                      14999 non-null  object
 6   retail_price             14942 non-null  float64
 7   discounted_price         14942 non-null  float64
 8   image                    14996 non-null  object
 9   is_FK_Advantage_product  14999 non-null  bool
 10  description              14998 non-null  object
 11  product_rating           14999 non-null  object
 12  overall_rating           14999 non-null  object
 13  brand                    10289 non-null  object
 14  product_specifications   14993 non-null  object
dtypes: bool(1), float64(2), object(12)
memory usage: 1.6+ MB
None
```

```
In [5]: # Display summary statistics of the dataset
        print(data.describe())
```

```
        retail_price   discounted_price
count   14942.000000   14942.000000
mean     3128.424040    2087.732298
std      9154.463342    7712.247869
min        35.000000      35.000000
25%       699.000000     349.000000
50%      1000.000000     499.000000
75%      1989.750000     999.000000
max    571230.000000  571230.000000
```

```
In [6]: print(data.isnull().sum())
```

```
uniq_id                     0
crawl_timestamp             0
product_url                 0
product_name                0
product_category_tree       0
pid                         0
retail_price               57
discounted_price           57
image                       3
is_FK_Advantage_product     0
description                 1
product_rating              0
overall_rating              0
brand                    4710
product_specifications      6
dtype: int64
```

```
In [5]:  sns.heatmap(data.isnull(), cbar=False, cmap='viridis')
         plt.title('Missing Values Heatmap')
         plt.show()
```


Missing Values Heatmap
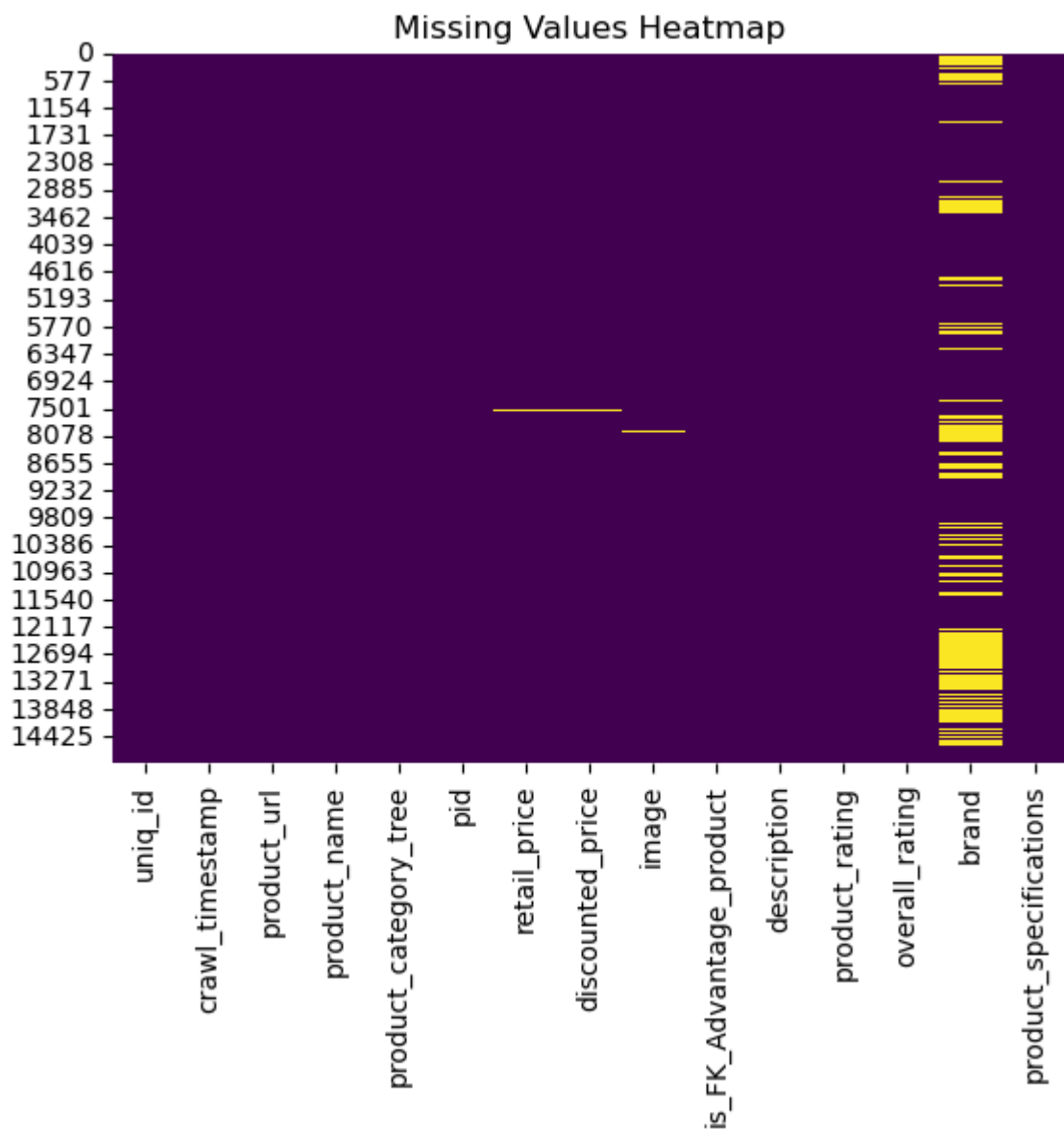
```
In [6]:  # Fill missing values in the 'brand' column with most used brand name
         most_common_brand = data['brand'].mode()[0]
         data['brand'] = data['brand'].fillna(most_common_brand)
```

```
In [8]:  # Drop rows where 'discription' has missing values
         data = data.dropna(subset=['description'])
```

```
In [9]: print(data.isnull().sum())
        print(f"The most common brand used for filling missing values is: {most_com
```

```
uniq_id                      0
crawl_timestamp              0
product_url                  0
product_name                 0
product_category_tree        0
pid                          0
retail_price                57
discounted_price            57
image                        3
is_FK_Advantage_product      0
description                  0
product_rating               0
overall_rating               0
brand                        0
product_specifications       6
dtype: int64
The most common brand used for filling missing values is: Allure Auto
```

```
In [10]: data['retail_price'].fillna(data['retail_price'].median(), inplace=True)
         data['discounted_price'].fillna(data['discounted_price'].median(), inplace=
```

```
In [11]: # Drop rows where 'image' and 'product specifications' column has missing v
         data = data.dropna(subset=['image'])
         data = data.dropna(subset=['product_specifications'])
```

```
In [12]: print(data.isnull().sum())
```

```
uniq_id                    0
crawl_timestamp            0
product_url                0
product_name               0
product_category_tree      0
pid                        0
retail_price               0
discounted_price           0
image                      0
is_FK_Advantage_product    0
description                0
product_rating             0
overall_rating             0
brand                      0
product_specifications     0
dtype: int64
```

```
In [13]:  # Check the distribution of product categories
          category_counts = data['product_category_tree'].value_counts()
          print(category_counts)
```

```
Clothing                    5312
Jewellery                   3218
Footwear                    1081
Automotive                   937
Mobiles & Accessories        767
Home Decor & Festive Needs   714
Kitchen & Dining             610
Computers                    531
Watches                      523
Tools & Hardware             379
Toys & School Supplies       271
Pens & Stationery            264
Baby Care                    224
Bags, Wallets & Belts        158
Name: product_category_tree, dtype: int64
```

```
In [14]:  plt.figure(figsize=(12, 6))
          sns.barplot(x=category_counts.index, y=category_counts.values, palette='vir
          plt.title('Distribution of Product Categories')
          plt.xlabel('Product Category')
          plt.ylabel('Frequency')
          plt.xticks(rotation=90)
          plt.show()
```
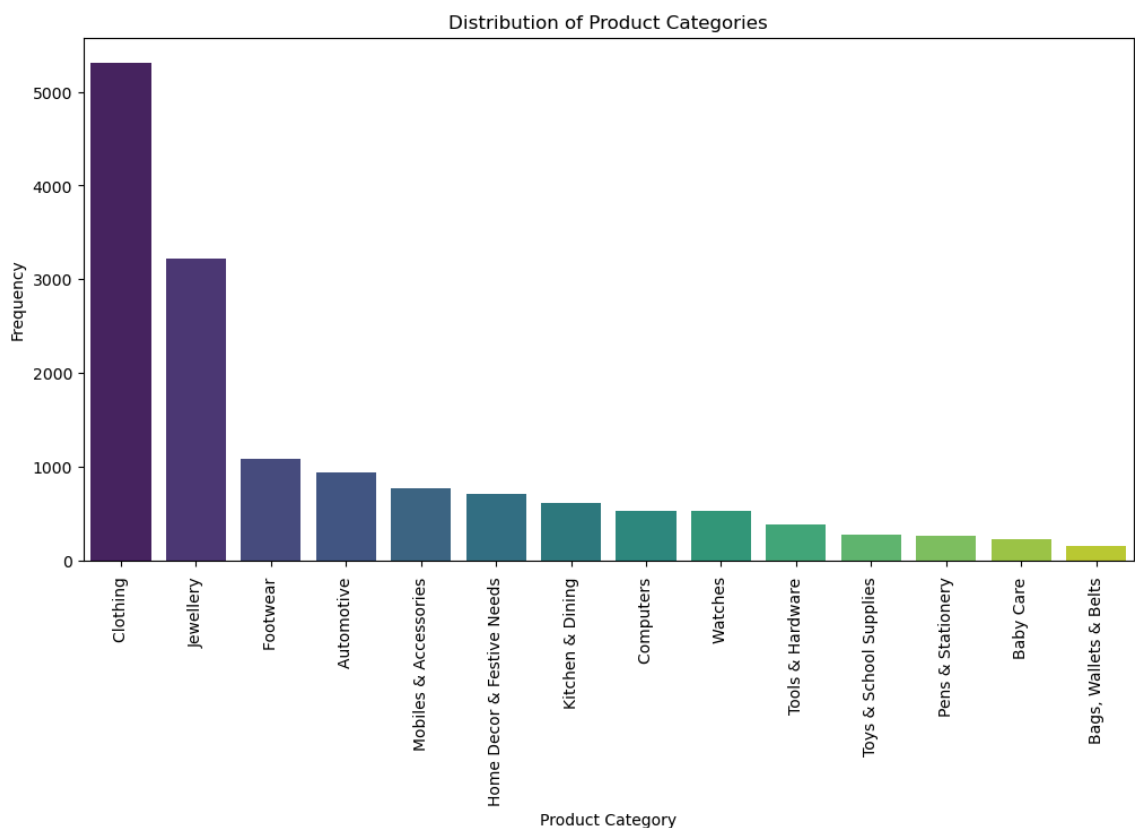
```
In [15]:  # Clean the text data
          def clean_text(text):
              text = re.sub(r'\W', ' ', text)  # Remove special characters
              text = re.sub(r'\s+', ' ', text)  # Remove extra whitespace
              text = text.lower()  # Convert to lowercase
              return text

          data['description'] = data['description'].apply(clean_text)
```

```
In [16]:  # Encode the target labels
          label_encoder = LabelEncoder()
          data['product_category_tree'] = label_encoder.fit_transform(data['product_c
```

```
In [17]:  # Split the data into training and validation sets
          X_train, X_val, y_train, y_val = train_test_split(data['description'], data
```

```
In [18]:  plt.figure(figsize=(12, 6))
          sns.countplot(y=data['product_category_tree'], order=data['product_category
          plt.title('Distribution of Product Categories')
          plt.show()
```
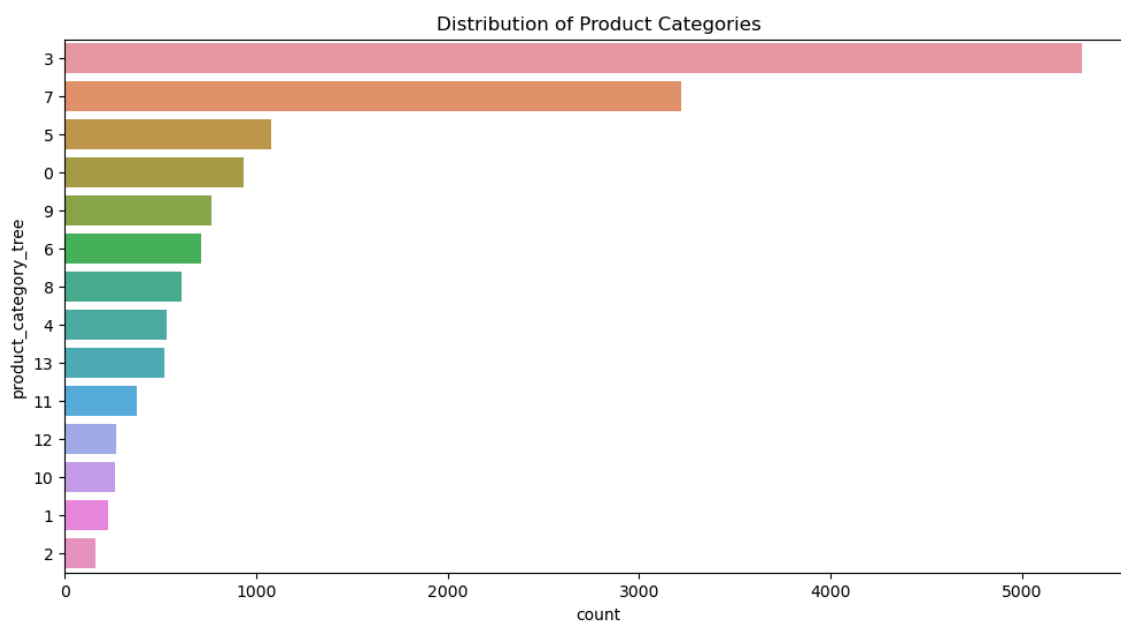
```
In [19]:  # Visualize the distribution of product ratings
          plt.figure(figsize=(12, 6))
          sns.histplot(data['product_rating'], bins=20)
          plt.title('Distribution of Product Ratings')
          plt.show()
```


Distribution of Product Ratings
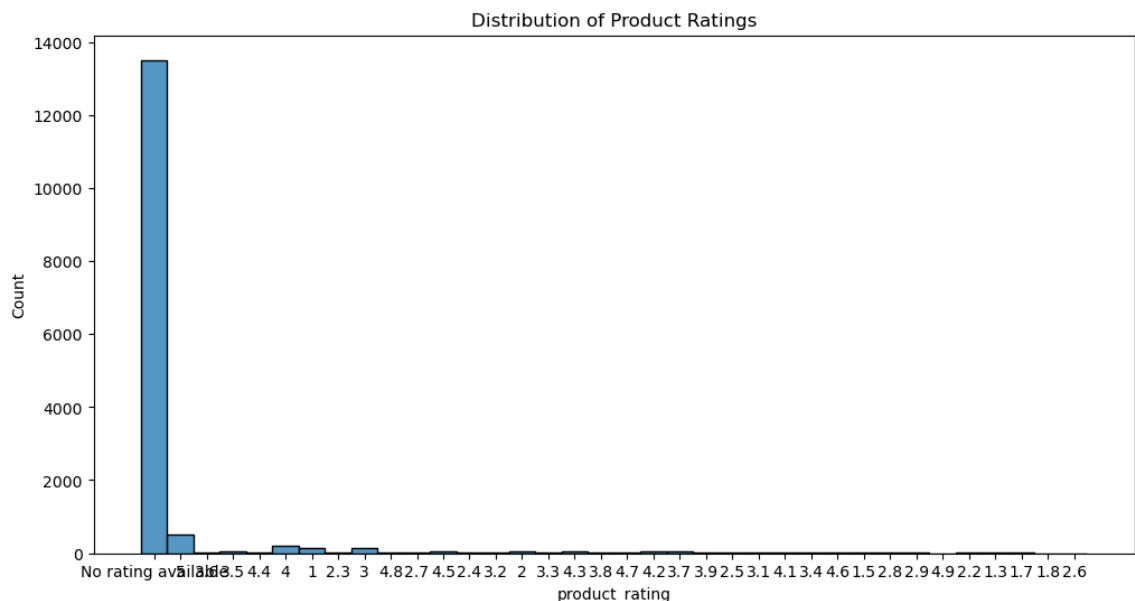
```
In [20]:  pip install wordcloud
```

Requirement already satisfied: wordcloud in c:\users\dell\anaconda3\lib\si
te-packages (1.9.3)
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\s
ite-packages (from wordcloud) (3.5.2)
Requirement already satisfied: numpy>=1.6.1 in c:\users\dell\anaconda3\lib
\site-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in c:\users\dell\anaconda3\lib\site-
packages (from wordcloud) (9.2.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda
3\lib\site-packages (from matplotlib->wordcloud) (1.4.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda
3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (20.9)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib
\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaco
nda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\sit
e-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```
In [21]:  from wordcloud import WordCloud
```

```
In [22]: wordcloud = WordCloud(width=800, height=400, background_color='white').gene
         plt.figure(figsize=(10, 5))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis('off')
         plt.title('Word Cloud for Product Descriptions')
         plt.show()
```


Word Cloud for Product Descriptions

```
In [23]: # Convert text data to TF-IDF features
         tfidf_vectorizer = TfidfVectorizer(max_features=5000)
         X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
         X_val_tfidf = tfidf_vectorizer.transform(X_val)
```

```
In [24]: from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import accuracy_score, classification_report

         # Train a Logistic Regression model
         lr_model = LogisticRegression(max_iter=1000)
         lr_model.fit(X_train_tfidf, y_train)
```

Out[24]: LogisticRegression(max_iter=1000)

```
In [25]:  # Predict on the validation set
          y_val_pred_lr = lr_model.predict(X_val_tfidf)

          # Evaluate the model
          print("Logistic Regression Accuracy:", accuracy_score(y_val, y_val_pred_lr)
          print("Logistic Regression Classification Report:\n", classification_report
```

```
Logistic Regression Accuracy: 0.9733155436957972
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       181
           1       0.97      0.74      0.84        46
           2       0.88      0.74      0.80        38
           3       0.97      1.00      0.99      1077
           4       0.98      0.94      0.96        88
           5       0.99      0.98      0.98       222
           6       0.93      1.00      0.96       127
           7       0.99      1.00      1.00       680
           8       0.97      0.96      0.97       114
           9       0.97      0.99      0.98       161
          10       0.96      0.49      0.65        47
          11       1.00      0.95      0.97        75
          12       0.71      0.81      0.76        42
          13       1.00      0.99      0.99       100

    accuracy                           0.97      2998
   macro avg       0.95      0.90      0.92      2998
weighted avg       0.97      0.97      0.97      2998
```

```python
In [26]:  from sklearn.ensemble import RandomForestClassifier
          from sklearn.metrics import accuracy_score, classification_report

          # Train a Random Forest model
          rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
          rf_model.fit(X_train_tfidf, y_train)

          # Predict on the validation set
          y_val_pred_rf = rf_model.predict(X_val_tfidf)

          # Evaluate the model
          print("Random Forest Accuracy:", accuracy_score(y_val, y_val_pred_rf))
          print("Random Forest Classification Report:\n", classification_report(y_val
```

```
Random Forest Accuracy: 0.9696464309539693
Random Forest Classification Report:
               precision    recall  f1-score   support

           0       0.99      0.98      0.99       181
           1       1.00      0.59      0.74        46
           2       0.91      0.79      0.85        38
           3       0.96      1.00      0.98      1077
           4       0.96      0.93      0.95        88
           5       1.00      0.94      0.97       222
           6       0.95      0.99      0.97       127
           7       0.99      1.00      0.99       680
           8       0.93      0.99      0.96       114
           9       0.97      0.96      0.96       161
          10       0.88      0.62      0.73        47
          11       0.99      0.96      0.97        75
          12       0.78      0.83      0.80        42
          13       1.00      0.97      0.98       100

    accuracy                           0.97      2998
   macro avg       0.95      0.90      0.92      2998
weighted avg       0.97      0.97      0.97      2998
```

```python
In [32]:  import tensorflow as tf
          from tensorflow.keras.models import Sequential
          from tensorflow.keras.layers import Dense, Dropout
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_9500\2839014353.py in <module>
----> 1 import tensorflow as tf
      2 from tensorflow.keras.models import Sequential
      3 from tensorflow.keras.layers import Dense, Dropout

ModuleNotFoundError: No module named 'tensorflow'
```

```
In [27]: pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\dell\anaconda3\lib\s
ite-packages (2.17.0)
Requirement already satisfied: tensorflow-intel==2.17.0 in c:\users\dell\a
naconda3\lib\site-packages (from tensorflow) (2.17.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\dell\anaconda3\li
b\site-packages (from tensorflow-intel==2.17.0->tensorflow) (1.14.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\dell\anacon
da3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (2.28.1)
Requirement already satisfied: packaging in c:\users\dell\anaconda3\lib\si
te-packages (from tensorflow-intel==2.17.0->tensorflow) (20.9)
Requirement already satisfied: h5py>=3.10.0 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow-intel==2.17.0->tensorflow) (3.11.0)
Requirement already satisfied: tensorboard<2.18,>=2.17 in c:\users\dell\an
aconda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (2.1
7.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\dell\anaconda3
\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (2.2.0)
Requirement already satisfied: six>=1.12.0 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow-intel==2.17.0->tensorflow) (1.16.0)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\dell\anaco
nda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (24.3.2
5)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.17.0->
tensorflow) (0.31.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\dell\anaconda3\l
ib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (2.1.0)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\s
ite-packages (from tensorflow-intel==2.17.0->tensorflow) (63.4.1)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\dell\a
naconda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (4.
5.0)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\u
sers\dell\anaconda3\lib\site-packages (from tensorflow-intel==2.17.0->tens
orflow) (0.6.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in c:\users\dell\an
aconda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (0.
4.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in c:\users\dell\anaco
nda3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (1.26.
4)
Requirement already satisfied: keras>=3.2.0 in c:\users\dell\anaconda3\lib
\site-packages (from tensorflow-intel==2.17.0->tensorflow) (3.4.1)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\dell\anaconda
3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (1.6.3)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\dell\anacon
da3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (0.2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\dell\anaconda
3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (3.3.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\dell\anacon
da3\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (1.65.4)
Requirement already satisfied: libclang>=13.0.0 in c:\users\dell\anaconda3
\lib\site-packages (from tensorflow-intel==2.17.0->tensorflow) (16.0.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.
3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in c:\users\dell\anaconda3\lib\site
-packages (from tensorflow-intel==2.17.0->tensorflow) (4.25.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\dell\anacond
a3\lib\site-packages (from astunparse>=1.6.0->tensorflow-intel==2.17.0->te
nsorflow) (0.37.1)
Requirement already satisfied: optree in c:\users\dell\anaconda3\lib\site-
packages (from keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (0.12.
```

```
1)
Requirement already satisfied: rich in c:\users\dell\anaconda3\lib\site-pa
ckages (from keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in c:\users\dell\anaconda3\lib\site-p
ackages (from keras>=3.2.0->tensorflow-intel==2.17.0->tensorflow) (0.0.8)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\dell\a
naconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.
17.0->tensorflow) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\anac
onda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.17.
0->tensorflow) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.17.0->tensor
flow) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anacond
a3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-intel==2.17.0->
tensorflow) (2022.9.14)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
c:\users\dell\anaconda3\lib\site-packages (from tensorboard<2.18,>=2.17->t
ensorflow-intel==2.17.0->tensorflow) (0.7.2)
Requirement already satisfied: markdown>=2.6.8 in c:\users\dell\anaconda3
\lib\site-packages (from tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0
->tensorflow) (3.3.4)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\dell\anaconda3
\lib\site-packages (from tensorboard<2.18,>=2.17->tensorflow-intel==2.17.0
->tensorflow) (2.0.3)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\dell\anaconda3
\lib\site-packages (from packaging->tensorflow-intel==2.17.0->tensorflow)
(3.0.9)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\dell\anac
onda3\lib\site-packages (from rich->keras>=3.2.0->tensorflow-intel==2.17.0
->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\dell\an
aconda3\lib\site-packages (from rich->keras>=3.2.0->tensorflow-intel==2.1
7.0->tensorflow) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\dell\anaconda3\lib\s
ite-packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow-i
ntel==2.17.0->tensorflow) (0.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [29]:
```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

In [31]:
```python
# Define the Deep Learning model
dl_model = Sequential()
dl_model.add(Dense(512, input_dim=X_train_tfidf.shape[1], activation='relu'
dl_model.add(Dropout(0.5))
dl_model.add(Dense(256, activation='relu'))
dl_model.add(Dropout(0.5))
dl_model.add(Dense(len(label_encoder.classes_), activation='softmax'))
```

```
In [32]:  # Compile the model
          dl_model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',
```

```
In [33]:  # Train the model
          dl_model.fit(X_train_tfidf.toarray(), y_train, epochs=10, batch_size=32, va
```

```
Epoch 1/10
375/375 ──────────────── 14s 33ms/step - accuracy: 0.6701 - loss: 1.16
07 - val_accuracy: 0.9746 - val_loss: 0.0836
Epoch 2/10
375/375 ──────────────── 12s 32ms/step - accuracy: 0.9798 - loss: 0.06
70 - val_accuracy: 0.9827 - val_loss: 0.0557
Epoch 3/10
375/375 ──────────────── 12s 32ms/step - accuracy: 0.9903 - loss: 0.03
09 - val_accuracy: 0.9817 - val_loss: 0.0538
Epoch 4/10
375/375 ──────────────── 12s 31ms/step - accuracy: 0.9946 - loss: 0.01
75 - val_accuracy: 0.9837 - val_loss: 0.0518
Epoch 5/10
375/375 ──────────────── 12s 32ms/step - accuracy: 0.9963 - loss: 0.01
34 - val_accuracy: 0.9863 - val_loss: 0.0544
Epoch 6/10
375/375 ──────────────── 11s 30ms/step - accuracy: 0.9971 - loss: 0.00
88 - val_accuracy: 0.9853 - val_loss: 0.0605
Epoch 7/10
375/375 ──────────────── 12s 32ms/step - accuracy: 0.9969 - loss: 0.00
86 - val_accuracy: 0.9833 - val_loss: 0.0705
Epoch 8/10
375/375 ──────────────── 13s 34ms/step - accuracy: 0.9968 - loss: 0.00
99 - val_accuracy: 0.9847 - val_loss: 0.0571
Epoch 9/10
375/375 ──────────────── 12s 31ms/step - accuracy: 0.9981 - loss: 0.00
53 - val_accuracy: 0.9833 - val_loss: 0.0704
Epoch 10/10
375/375 ──────────────── 11s 30ms/step - accuracy: 0.9985 - loss: 0.00
44 - val_accuracy: 0.9830 - val_loss: 0.0727
```

Out[33]: <keras.src.callbacks.history.History at 0x1f3729009d0>

```
In [34]:  # Predict on the validation set
          y_val_pred_dl = np.argmax(dl_model.predict(X_val_tfidf.toarray()), axis=1)
```

```
94/94 ──────────────── 0s 3ms/step
```

```
In [35]:  # Evaluate the model
          print("Deep Learning Model Accuracy:", accuracy_score(y_val, y_val_pred_dl)
          print("Deep Learning Model Classification Report:\n", classification_report
```

```
Deep Learning Model Accuracy: 0.9829886591060707
Deep Learning Model Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.96      0.97       181
           1       0.97      0.85      0.91        46
           2       0.89      0.89      0.89        38
           3       0.99      1.00      1.00      1077
           4       0.90      0.97      0.93        88
           5       1.00      0.98      0.99       222
           6       0.95      0.99      0.97       127
           7       1.00      1.00      1.00       680
           8       0.99      0.97      0.98       114
           9       0.98      0.96      0.97       161
          10       0.88      0.81      0.84        47
          11       0.97      1.00      0.99        75
          12       0.80      0.88      0.84        42
          13       1.00      1.00      1.00       100

    accuracy                           0.98      2998
   macro avg       0.95      0.95      0.95      2998
weighted avg       0.98      0.98      0.98      2998
```

```
In [36]:  #fine tuning
          from sklearn.model_selection import GridSearchCV

          param_grid = {
              'C': [0.1, 1, 10],
              'solver': ['liblinear', 'lbfgs']
          }

          grid_search = GridSearchCV(LogisticRegression(max_iter=1000), param_grid, c
          grid_search.fit(X_train_tfidf, y_train)

          print("Best parameters found: ", grid_search.best_params_)
          print("Best accuracy found: ", grid_search.best_score_)
```

```
Best parameters found:  {'C': 10, 'solver': 'lbfgs'}
Best accuracy found:  0.9801517243249463
```

```
In [37]:  test_data = pd.read_csv(r'C:\Users\DELL\Downloads\test_data.csv')
```

```
In [39]:  test_results = pd.read_excel(r'C:\Users\DELL\Downloads\test_results.xlsx')
```

```
In [40]: print(test_data.head())
         print(test_results.head())
```

```
                              uniq_id          crawl_timestamp  \
0  4fb99d98225f415e7ece96938e95628f  2015-12-20 08:26:17 +0000
1  4ea284c8d38b2ea97a1c2a26f34e057c  2015-12-20 08:26:17 +0000
2  ee6ce2c7045c54257e2a0b590e09c296  2015-12-20 08:26:17 +0000
3  e797ba3b5f2e2d1fdc520e48486ab60e  2015-12-20 08:26:17 +0000
4  f4d8d43858c8858c68d75ce07ac641c0  2015-12-20 08:26:17 +0000

                                         product_url  \
0  http://www.flipkart.com/v-v-art-brass-bracelet... (http://www.flipkart.
com/v-v-art-brass-bracelet...)
1  http://www.flipkart.com/kalpaveda-copper-cuff/... (http://www.flipkart.
com/kalpaveda-copper-cuff/...)
2  http://www.flipkart.com/thelostpuppy-book-cove... (http://www.flipkart.
com/thelostpuppy-book-cove...)
3  http://www.flipkart.com/riana-copper-bangle/p/... (http://www.flipkart.
com/riana-copper-bangle/p/...)
4  http://www.flipkart.com/inox-jewelry-stainless... (http://www.flipkart.
com/inox-jewelry-stainless...)

                        product_name             pid  retail_pr
ice  \
0              V&V ART Brass Bracelet  BBAE6NYHCDTEZJTB         47
0.0
1         Kalpaveda Copper Copper Cuff  BBAEDFFKZJTY7SZZ        120
0.0
2  Thelostpuppy Book Cover for Apple iPad Air  ACCEA4DZH6M5SFVH        219
9.0
3          Riana Copper Copper Bangle  BBAEAXFQHHMF3EYZ         249
9.0
4      Inox Jewelry Stainless Steel Cuff  BBAECH63WYDG6TE2        162
9.0

   discounted_price                                      image  \
0            423.0  ["http://img6a.flixcart.com/image/bangle-brace...
1           1200.0  ["http://img6a.flixcart.com/image/bangle-brace...
2            599.0  ["http://img5a.flixcart.com/image/cases-covers...
3            649.0  ["http://img5a.flixcart.com/image/bangle-brace...
4           1222.0  ["http://img6a.flixcart.com/image/bangle-brace...

   is_FK_Advantage_product                                    descript
ion  \
0                    False  V&V ART Brass Bracelet - Buy V&V ART Brass Br
a...
1                    False  Kalpaveda Copper Copper Cuff\n
...
2                    False  Thelostpuppy Book Cover for Apple iPad Air (M
u...
3                    False  Riana Copper Copper Bangle - Buy Riana Copper
...
4                    False   Inox Jewelry Stainless Steel Cuff\n
...

         product_rating        overall_rating         brand  \
0  No rating available  No rating available       V&V ART
1  No rating available  No rating available     Kalpaveda
2  No rating available  No rating available  Thelostpuppy
3                    5                    5         Riana
4  No rating available  No rating available   Inox Jewelry

                        product_specifications
0  {"product_specification"=>[{"key"=>"Brand", "v...
```

```
1  {"product_specification"=>[{"key"=>"Stretchabl...
2  {"product_specification"=>[{"key"=>"Brand", "v...
3  {"product_specification"=>[{"key"=>"Collection...
4  {"product_specification"=>[{"key"=>"Stretchabl...
   product_category_tree
0              Jewellery
1              Jewellery
2  Mobiles & Accessories
3              Jewellery
4              Jewellery
```

In [41]:
```python
test_data['description'] = test_data['description'].fillna('')
test_data['description'] = test_data['description'].apply(clean_text)
X_test_tfidf = tfidf_vectorizer.transform(test_data['description'])
```

In [42]:
```python
# Predict using Logistic Regression
y_test_pred_lr = lr_model.predict(X_test_tfidf)
```

In [43]:
```python
# Predict using Random Forest
y_test_pred_rf = rf_model.predict(X_test_tfidf)
```

In [44]:
```python
# Predict using Deep Learning model
y_test_pred_dl = np.argmax(dl_model.predict(X_test_tfidf.toarray()), axis=1
```

```
80/80 ──────────────────── 0s 3ms/step
```

In [48]:
```python
# Preprocess the description column for test data
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_test_tfidf = tfidf_vectorizer.fit_transform(test_data['description'])
```

In [50]:
```python
label_encoder = LabelEncoder()
label_encoder.fit(data['product_category_tree'])
```

Out[50]: LabelEncoder()

In [55]:
```python
# Ensure all labels are strings
data['product_category_tree'] = data['product_category_tree'].astype(str)
test_results['product_category_tree'] = test_results['product_category_tree
```

In [57]:
```python
# Preprocess the description column for test data (assuming TF-IDF)
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_test_tfidf = tfidf_vectorizer.fit_transform(test_data['description'])
```

In [59]:
```python
# Combine all labels from train and test to re-fit LabelEncoder
combined_labels = pd.concat([data['product_category_tree'], test_results['p
label_encoder = LabelEncoder()
label_encoder.fit(combined_labels)
```

Out[59]: LabelEncoder()

```python
In [60]: # Encode the true test labels
         y_test_true_encoded = label_encoder.transform(test_results['product_category'
```

```python
In [61]: # Predict with Logistic Regression model
         y_test_pred_lr = lr_model.predict(X_test_tfidf)
```

```
In [64]: print("Logistic Regression Test Accuracy:", accuracy_score(y_test_true_enco
         print("Logistic Regression Test Classification Report:\n", classification_r
```

```
Logistic Regression Test Accuracy: 0.0
Logistic Regression Test Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       0.0
           3       0.00      0.00      0.00       0.0
           4       0.00      0.00      0.00       0.0
           5       0.00      0.00      0.00       0.0
           6       0.00      0.00      0.00       0.0
           7       0.00      0.00      0.00       0.0
           8       0.00      0.00      0.00       0.0
           9       0.00      0.00      0.00       0.0
          12       0.00      0.00      0.00       0.0
          14       0.00      0.00      0.00      75.0
          15       0.00      0.00      0.00     259.0
          16       0.00      0.00      0.00     107.0
          17       0.00      0.00      0.00     882.0
          18       0.00      0.00      0.00      47.0
          19       0.00      0.00      0.00     144.0
          20       0.00      0.00      0.00     215.0
          21       0.00      0.00      0.00     313.0
          22       0.00      0.00      0.00      37.0
          23       0.00      0.00      0.00     331.0
          24       0.00      0.00      0.00      49.0
          25       0.00      0.00      0.00      12.0
          26       0.00      0.00      0.00      59.0
          27       0.00      0.00      0.00       4.0

    accuracy                           0.00    2534.0
   macro avg       0.00      0.00      0.00    2534.0
weighted avg       0.00      0.00      0.00    2534.0
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [65]:
```python
# Predict with Random Forest model
y_test_pred_rf = rf_model.predict(X_test_tfidf)
```

```
In [67]:  #Evaluate Random Forest model
          print("Random Forest Test Accuracy:", accuracy_score(y_test_true_encoded, y
          print("Random Forest Test Classification Report:\n", classification_report(
```

```
Random Forest Test Accuracy: 0.0
Random Forest Test Classification Report:
               precision    recall  f1-score   support

           0       0.00      0.00      0.00       0.0
           3       0.00      0.00      0.00       0.0
           4       0.00      0.00      0.00       0.0
           5       0.00      0.00      0.00       0.0
           6       0.00      0.00      0.00       0.0
           7       0.00      0.00      0.00       0.0
           9       0.00      0.00      0.00       0.0
          14       0.00      0.00      0.00      75.0
          15       0.00      0.00      0.00     259.0
          16       0.00      0.00      0.00     107.0
          17       0.00      0.00      0.00     882.0
          18       0.00      0.00      0.00      47.0
          19       0.00      0.00      0.00     144.0
          20       0.00      0.00      0.00     215.0
          21       0.00      0.00      0.00     313.0
          22       0.00      0.00      0.00      37.0
          23       0.00      0.00      0.00     331.0
          24       0.00      0.00      0.00      49.0
          25       0.00      0.00      0.00      12.0
          26       0.00      0.00      0.00      59.0
          27       0.00      0.00      0.00       4.0

    accuracy                           0.00    2534.0
   macro avg       0.00      0.00      0.00    2534.0
weighted avg       0.00      0.00      0.00    2534.0
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [68]:
```python
# Predict with Deep Learning model (assuming you convert sparse matrix to a
y_test_pred_dl = np.argmax(dl_model.predict(X_test_tfidf.toarray()), axis=1
```

80/80 ──────────────── 0s 3ms/step

```
# Evaluate Deep Learning model
print("Deep Learning Model Test Accuracy:", accuracy_score(y_test_true_enco
print("Deep Learning Model Test Classification Report:\n", classification_r
```

Deep Learning Model Test Accuracy: 0.0
Deep Learning Model Test Classification Report:
              precision   recall f1-score   support

           0   0.00   0.00   0.00   0.0
           1   0.00   0.00   0.00   0.0
           2   0.00   0.00   0.00   0.0
           3   0.00   0.00   0.00   0.0
           4   0.00   0.00   0.00   0.0
           5   0.00   0.00   0.00   0.0
           6   0.00   0.00   0.00   0.0
           7   0.00   0.00   0.00   0.0
           8   0.00   0.00   0.00   0.0
           9   0.00   0.00   0.00   0.0
          10   0.00   0.00   0.00   0.0
          11   0.00   0.00   0.00   0.0
          12   0.00   0.00   0.00   0.0
          13   0.00   0.00   0.00   0.0
          14   0.00   0.00   0.00   75.0
          15   0.00   0.00   0.00   259.0
          16   0.00   0.00   0.00   107.0
          17   0.00   0.00   0.00   882.0
          18   0.00   0.00   0.00   47.0
          19   0.00   0.00   0.00   144.0
          20   0.00   0.00   0.00   215.0
          21   0.00   0.00   0.00   313.0
          22   0.00   0.00   0.00   37.0
          23   0.00   0.00   0.00   331.0
          24   0.00   0.00   0.00   49.0
          25   0.00   0.00   0.00   12.0
          26   0.00   0.00   0.00   59.0
          27   0.00   0.00   0.00   4.0

    accuracy   0.00   2534.0
   macro avg   0.00   0.00   0.00   2534.0
weighted avg   0.00   0.00   0.00   2534.0
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\metrics\_classification.
py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and be
ing set to 0.0 in labels with no true samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

```
from sklearn.metrics import accuracy_score, classification_report

# Check some predictions and true values
print("Sample True Labels:", y_test_true_encoded[:10])
print("Sample Logistic Regression Predictions:", y_test_pred_lr[:10])

# Evaluate Logistic Regression model
print("Logistic Regression Test Accuracy:", accuracy_score(y_test_true_enco
print("Logistic Regression Test Classification Report:\n", classification_r

# Evaluate Random Forest model
print("Random Forest Test Accuracy:", accuracy_score(y_test_true_encoded, y
print("Random Forest Test Classification Report:\n", classification_report(
```

```
Sample True Labels: [21 21 23 21 21 23 21 23 21 23]
Sample Logistic Regression Predictions: [3 3 3 3 3 3 3 3 3 3]
Logistic Regression Test Accuracy: 0.0
Logistic Regression Test Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       0.0
           3       0.00      0.00      0.00       0.0
           4       0.00      0.00      0.00       0.0
           5       0.00      0.00      0.00       0.0
           6       0.00      0.00      0.00       0.0
           7       0.00      0.00      0.00       0.0
           8       0.00      0.00      0.00       0.0
           9       0.00      0.00      0.00       0.0
          12       0.00      0.00      0.00       0.0
          14       0.00      0.00      0.00      75.0
          15       0.00      0.00      0.00     259.0
          16       0.00      0.00      0.00     107.0
          17       0.00      0.00      0.00     882.0
```