Unsupported Cell Type. Double-Click to inspect/edit the content.

```
#Preprocessing the Dataset:
```

```
import pandas as pd
import re
import string

# Load the dataset
df = pd.read_csv('tripadvisor_hotel_reviews.csv')
#working with 1000 rows
df=df.iloc[:1000]

# Preprocess the text column
df['Review'] = df['Review'].str.lower()
df['Review'] = df['Review'].str.replace('[^\w\s]', '', regex=True)
df['Review'] = df['Review'].str.replace('\d+', '', regex=True)
df['Review'] = df['Review'].str.strip()
```

```
df['lowered']=df['Review']
```

```
df
```

	Review	Rating	lowered
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...
4	great stay great stay went seahawk game awesom...	5	great stay great stay went seahawk game awesom...
...	...	...	...
995	average price hotel good location blocks unio...	3	average price hotel good location blocks unio...
996	good customer service recently wrote dissatisf...	4	good customer service recently wrote dissatisf...

```
#Start by importing the NLTK library, which contains a list of stop words.
import nltk
#Next, you can create a list of stop words by calling the nltk.corpus.stopwords.words() function and passing the language
#you want to use. For example, to create a list of English stop words, you can do:
stop_words = nltk.corpus.stopwords.words('english')
print(stop_words)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yours
```

```
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

# Define the stemmer to be used
stemmer = PorterStemmer()

def preprocess_text(text):
    # Tokenize the text
    words = word_tokenize(text)

    # Remove stop words
    words = [word for word in words if word.lower() not in stop_words]

    # Perform stemming on the remaining words
    stemmed_words = [stemmer.stem(word) for word in words]

    return stemmed_words
```

```
# Preprocess the text in the dataset
df["processed_text"] = df["lowered"].apply(preprocess_text)
```

```
df
```

	Review	Rating	lowered	processed_text
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...	[nice, hotel, expens, park, got, good, deal, s...
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...	[ok, noth, special, charg, diamond, member, hi...
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...	[nice, room, experi, hotel, monaco, seattl, go...
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...	[uniqu, great, stay, wonder, time, hotel, mona...
4	great stay great stay went seahawk game awesom...	5	great stay great stay went seahawk game awesom...	[great, stay, great, stay, went, seahawk, game...
...	...	...	...	...
495	questionable quality property wonderful ways r...	3	questionable quality property wonderful ways r...	[question, qualiti, properti, wonder, way, req...
496	luxurious pricey stay hilton purchased rooms h...	3	luxurious pricey stay hilton purchased rooms h...	[luxuri, pricey, stay, hilton, purchas, room, ...

```
#Identify the aspects in the text:
```

```
!pip install spacy
!python -m spacy download en_core_web_sm
```

```
    Traceback (most recent call last):
      File "C:\Users\Yugendra\anaconda3\lib\site-packages\requests\adapters.py", line 439, in send
        resp = conn.urlopen(
      File "C:\Users\Yugendra\anaconda3\lib\site-packages\urllib3\connectionpool.py", line 755, in urlopen
        retries = retries.increment(
      File "C:\Users\Yugendra\anaconda3\lib\site-packages\urllib3\util\retry.py", line 532, in increment
        raise six.reraise(type(error), error, _stacktrace)
      File "C:\Users\Yugendra\anaconda3\lib\site-packages\urllib3\packages\six.py", line 734, in reraise
        raise value.with_traceback(tb)
      File "C:\Users\Yugendra\anaconda3\lib\site-packages\urllib3\connectionpool.py", line 699, in urlopen
    File "C:\Users\Yugendra\anaconda3\lib\site-packages\requests\sessions.py", line 655, in send
      r = adapter.send(request, **kwargs)
    File "C:\Users\Yugendra\anaconda3\lib\site-packages\requests\adapters.py", line 498, in send
      raise ConnectionError(err, request=request)
  requests.exceptions.ConnectionError: ('Connection aborted.', ConnectionResetError(10054, 'An existing connection was forcibly cl
```

```
import spacy

# Load the NER model
nlp = spacy.load("en_core_web_sm")

# Function to extract the entities from the text
def extract_entities(text):
    text_str = ' '.join(text)
    doc = nlp(text_str)
    entities = [ent.text for ent in doc.ents]
    return entities


# Extract the entities from the text column
df['aspects'] = df['processed_text'].apply(extract_entities)
```

```
df
```

	Review	Rating	lowered	processed_text	aspects
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...	[nice, hotel, expens, park, got, good, deal, s...	[anniversari arriv, advic previou, valet park,...
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...	[ok, noth, special, charg, diamond, member, hi...	[kimpton websit, kimpton call, kimpton prefer,...
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...	[nice, room, experi, hotel, monaco, seattl, go...	[separ wakeup, mediterranean suit, weekend, ni...
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...	[uniqu, great, stay, wonder, time, hotel, mona...	[brandi, featur great staff friendli, palatt a...
4	great stay great stay went seahawk game awesom...	5	great stay great stay went seahawk game awesom...	[great, stay, great, stay, went, seahawk, game...	[english, arab, second, larg citi, compani, bu...
...	...	...	...	...	...
995	average price hotel good location blocks unio...	3	average price hotel good location blocks unio...	[averag, price, hotel, good, locat, block, uni...	[servic adequ room]
996	good customer service recently wrote dissatisf...	4	good customer service recently wrote dissatisf...	[good, custom, servic, recent, wrote, dissatis...	[new year eve, parti speak, manag rd partyin a...
997	ignore bad press just post review reading nega...	4	ignore bad press just post review reading nega...	[ignor, bad, press, post, review, read, neg, c...	[ignor bad press post review, neg comment post...
998	business trip ok hotel fine evening business m...	3	business trip ok hotel fine evening business m...	[busi, trip, ok, hotel, fine, even, busi, matt...	[busi trip, busi mattress comfort room, kudo b...
999	okay downstairs medicore room overrated overpr...	3	okay downstairs medicore room overrated overpr...	[okay, downstair, medicor, room, overr, overpr...	[downstair medicor, overr, european, unflatt b...

1000 rows × 5 columns

```
# Filter the dataframe to keep only the reviews with aspects
df_with_aspects = df[df['aspects'].apply(lambda x: len(x) > 0)]
```

```
df_with_aspects
```

	Review	Rating	lowered	processed_text	aspects
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...	[nice, hotel, expens, park, got, good, deal, s...	[anniversari arriv, advic previou, valet park,...
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...	[ok, noth, special, charg, diamond, member, hi...	[kimpton websit, kimpton call, kimpton prefer,...
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...	[nice, room, experi, hotel, monaco, seattl, go...	[separ wakeup, mediterranean suit, weekend, ni...
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...	[uniqu, great, stay, wonder, time, hotel, mona...	[brandi, featur great staff friendli, palatt a...
	great stay great		great stay great		
...	...	...	...	...	...

```
# Filter the dataframe to keep only the reviews without aspects
df_without_aspects = df[df['aspects'].apply(lambda x: len(x) == 0)]
```

```
df_without_aspects
```

[illegible]	stayed nights business trip gr...	5	stayed nights business trip gr...	night, busi, trip, g...	[]
461	doubletree ok no problems great view airport r...	4	doubletree ok no problems great view airport r...	[doubletre, ok, problem, great, view, airport,...	[]
466	great deal come stayed hotel day trip vancouv...	5	great deal come stayed hotel day trip vancouv...	[great, deal, come, stay, hotel, day, trip, va...	[]
467	comfortable pleasant stay stayed hotel nights ...	4	comfortable pleasant stay stayed hotel nights ...	[comfort, pleasant, stay, stay, hotel, night, ...	[]
478	usual best hotel clean managed tired traveller...	5	usual best hotel clean managed tired traveller...	[usual, best, hotel, clean, manag, tire, trave...	[]
484	great experience originally went reservation b...	4	great experience originally went reservation b...	[great, experi, origin, went, reserv, bit, war...	[]
491	nice comfortable rooms comfortable bed awesome...	4	nice comfortable rooms comfortable bed awesome...	[nice, comfort, room, comfort, bed, awesom, at...	[]
501	loved hotel pleasant surprise booked travel of...	4	loved hotel pleasant surprise booked travel of...	[love, hotel, pleasant, surpris, book, travel,...	[]
527	globus tour stayed globus tour tram bustrain s...	4	globus tour stayed globus tour tram bustrain s...	[globu, tour, stay, globu, tour, tram, bustrai...	[]
529	pink black funky modern building great locatio...	4	pink black funky modern building great locatio...	[pink, black, funki, modern, build, great, loc...	[]
531	great hotel dont drink bottled water modern	4	great hotel dont drink bottled water modern	[great, hotel, dont, drink, bottl, water,	[]
[illegible]	extra fees room nice quiet extra fees high	[illegible]	extra fees room nice quiet extra fees high	[extra, fee, room, nice,	[]

```
df=df_with_aspects
```

```
df
```

	Review	Rating	lowered	processed_text	aspects
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...	[nice, hotel, expens, park, got, good, deal, s...	[anniversari arriv, advic previou, valet park,...
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...	[ok, noth, special, charg, diamond, member, hi...	[kimpton websit, kimpton call, kimpton prefer,...
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...	[nice, room, experi, hotel, monaco, seattl, go...	[separ wakeup, mediterranean suit, weekend, ni...
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...	[uniqu, great, stay, wonder, time, hotel, mona...	[brandi, featur great staff friendli, palatt a...
4	great stay great stay went seahawk game awesom...	5	great stay great stay went seahawk game awesom...	[great, stay, great, stay, went, seahawk, game...	[english, arab, second, larg citi, compani, bu...
...	...	...	...	...	...
995	average price hotel good location blocks unio...	3	average price hotel good location blocks unio...	[averag, price, hotel, good, locat, block, uni...	[servic adequ room]
996	good customer service recently wrote dissatisf...	4	good customer service recently wrote dissatisf...	[good, custom, servic, recent, wrote, dissatis...	[new year eve, parti speak, manag rd partyin a...

```
!pip install transformers
```

```
    Requirement already satisfied: transformers in c:\users\yugendra\anaconda3\lib\site-packages (4.26.1)
    Requirement already satisfied: numpy>=1.17 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (1.20.1)
    Requirement already satisfied: packaging>=20.0 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (20.9)
    Requirement already satisfied: tqdm>=4.27 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (4.59.0)
    Requirement already satisfied: pyyaml>=5.1 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (5.4.1)
    Requirement already satisfied: regex!=2019.12.17 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (2021.4.4)
    Requirement already satisfied: requests in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (2.25.1)
    Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (0
Requirement already satisfied: filelock in c:\users\yugendra\anaconda3\lib\site-packages (from transformers) (3.0.12)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in c:\users\yugendra\anaconda3\lib\site-packages (from transformer
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\yugendra\anaconda3\lib\site-packages (from huggingface-hub<1.
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\yugendra\anaconda3\lib\site-packages (from packaging>=20.0->transformer
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\yugendra\anaconda3\lib\site-packages (from requests->transformers)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\yugendra\anaconda3\lib\site-packages (from requests->transformers) (4.
Requirement already satisfied: idna<3,>=2.5 in c:\users\yugendra\anaconda3\lib\site-packages (from requests->transformers) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\yugendra\anaconda3\lib\site-packages (from requests->transformers) (2
```

```
!pip install git+https://github.com/huggingface/transformers
```

```
Collecting git+https://github.com/huggingface/transformers
  Cloning https://github.com/huggingface/transformers to c:\users\yugendra\appdata\local\temp\pip-req-build-05ozbejh
  Running command git clone -q https://github.com/huggingface/transformers 'C:\Users\Yugendra\AppData\Local\Temp\pip-req-build-05oz
  ERROR: Error [WinError 2] The system cannot find the file specified while executing command git clone -q https://github.com/huggi
ERROR: Cannot find command 'git' - do you have 'git' installed and in your PATH?
```

```
import pandas as pd
import torch
import transformers

# Load the pre-trained tokenizer
tokenizer = transformers.BertTokenizer.from_pretrained('bert-base-uncased')

# Load the pre-trained sentiment classifier
sentiment_classifier = transformers.AutoModelForSequenceClassification.from_pretrained("nlptown/bert-base-multilingual-uncased-sentiment"

# Define a function to perform sentiment analysis
def predict_sentiment(text):
    # Tokenize the text data
    inputs = tokenizer(text, return_tensors='pt')
    # Predict the sentiment of the text data
    sentiment = sentiment_classifier(**inputs).logits
    # Return the predicted sentiment as an integer
    return torch.argmax(sentiment).item()

# Predict the sentiment of each aspect in the text data
df['sentiments'] = df['aspects'].apply(lambda x: [predict_sentiment(text) for text in x])

# Tokenize the text data
df['tokenized_aspects'] = df['aspects'].apply(lambda x: [tokenizer.tokenize(text) for text in x])

# Convert tokenized text data to numerical values
df['encoded_aspects'] = df['tokenized_aspects'].apply(lambda x: [tokenizer.convert_tokens_to_ids(tokens) for tokens in x])
```

```
<ipython-input-17-e3c1df07496a>:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['sentiments'] = df['aspects'].apply(lambda x: [predict_sentiment(text) for text in x])
<ipython-input-17-e3c1df07496a>:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['tokenized_aspects'] = df['aspects'].apply(lambda x: [tokenizer.tokenize(text) for text in x])
<ipython-input-17-e3c1df07496a>:27: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['encoded_aspects'] = df['tokenized_aspects'].apply(lambda x: [tokenizer.convert_tokens_to_ids(tokens) for tokens in x])
```

```
# Convert encoded aspects to padded sequences
df['padded_aspects'] = df['encoded_aspects'].apply(lambda x: torch.nn.utils.rnn.pad_sequence([torch.tensor(aspect) for aspect in x], padd

# Convert sentiments to a numerical format
df['labels'] = df['sentiments'].apply(lambda x: [1 if sentiment > 0 else 0 for sentiment in x])
```

```
<ipython-input-18-baa3284c34e1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
```

```
  df['padded_aspects'] = df['encoded_aspects'].apply(lambda x: torch.nn.utils.rnn.pad_sequence([torch.tensor(aspect) for aspect in
<ipython-input-18-baa3284c34e1>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df['labels'] = df['sentiments'].apply(lambda x: [1 if sentiment > 0 else 0 for sentiment in x])
```

```
df
```

	Review	Rating	lowered	processed_text	aspects	sentiments	token
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking got good deal sta...	[nice, hotel, expens, park, got, good, deal, s...	[anniversari arriv, advic previou, valet park,...	[4, 3, 3, 2, 4]	[[ann ar,
1	ok nothing special charge diamond member hilto...	2	ok nothing special charge diamond member hilto...	[ok, noth, special, charg, diamond, member, hi...	[kimpton websit, kimpton call, kimpton prefer,...	[4, 2, 2, 0, 3, 2, 0, 2, 2, 2, 2, 4, 2, 2, 0, ...	[[kin ##sit
2	nice rooms not experience hotel monaco seattl...	3	nice rooms not experience hotel monaco seattl...	[nice, room, experi, hotel, monaco, seattl, go...	[separ wakeup, mediterranean suit, weekend, ni...	[0, 2, 2, 3, 3, 1, 2, 2, 3, 3]	[[s [
3	unique great stay wonderful time hotel monaco ...	5	unique great stay wonderful time hotel monaco ...	[uniqu, great, stay, wonder, time, hotel, mona...	[brandi, featur great staff friendli, palatt a...	[2, 2, 4]	[[br ##
4	great stay great stay went seahawk game awesom...	5	great stay great stay went seahawk game awesom...	[great, stay, great, stay, went, seahawk, game...	[english, arab, second, larg citi, compani, bu...	[4, 2, 3, 4, 2, 0, 0, 4, 0]	[[e [sec
...	...	...	...	...	...	...	
995	average price hotel good location blocks unio...	3	average price hotel good location blocks unio...	[averag, price, hotel, good, locat, block, uni...	[servic adequ room]	[3]	[[ser,
996	good customer service recently wrote dissatisf...	4	good customer service recently wrote dissatisf...	[good, custom, servic, recent, wrote, dissatis...	[new year eve, parti speak, manag rd partyin a...	[0, 0, 0, 2, 2]	[[n [pa
997	ignore bad press just post review reading nega...	4	ignore bad press just post review reading nega...	[ignor, bad, press, post, review, read, neg, c...	[ignor bad press post review, neg comment post...	[0, 0, 4, 1, 3, 3, 2, 3, 2, 2, 2]	[press

```
# Set the maximum sequence length
MAX_LEN = 128

# Create the input ids and attention masks for each aspect
input_ids = []
attention_masks = []

# Tokenize each aspect in the text
```

```
for encoded_aspect in df['encoded_aspects'].values:
    aspect_input_ids = []
    aspect_attention_masks = []
    for aspect in encoded_aspect:
        # Convert the aspect to input ids and attention masks
        encoded_aspect = tokenizer.encode(aspect, add_special_tokens=True, max_length=MAX_LEN, truncation=True)
        aspect_input_ids.append(encoded_aspect)
        aspect_attention_mask = [1] * len(encoded_aspect)
        aspect_attention_masks.append(aspect_attention_mask)

    # Pad the input ids and attention masks to the maximum length
    for i in range(len(aspect_input_ids)):
        padding_length = MAX_LEN - len(aspect_input_ids[i])
        aspect_input_ids[i] = aspect_input_ids[i] + [0] * padding_length
        aspect_attention_masks[i] = aspect_attention_masks[i] + [0] * padding_length

    # Append each aspect's input ids and attention masks to the lists
    input_ids.append(aspect_input_ids)
    attention_masks.append(aspect_attention_masks)



# Determine the maximum number of aspects for any text
max_num_aspects = max([len(ids) for ids in input_ids])

# Pad the input_ids and attention_masks lists to the maximum number of aspects
for i in range(len(input_ids)):
    padding_length = max_num_aspects - len(input_ids[i])
    # Append [0] * MAX_LEN for the required number of times to pad the input_ids and attention_masks
    input_ids[i] += [[0] * MAX_LEN] * padding_length
    attention_masks[i] += [[0] * MAX_LEN] * padding_length

# Convert the input_ids and attention_masks lists to tensors
input_ids = torch.tensor(input_ids)
attention_masks = torch.tensor(attention_masks)


# Initialize a list to store the hidden states
all_last_hidden_states = []

# Extract the contextual embeddings from the text data
with torch.no_grad():
    for i in range(input_ids.shape[0]):
        last_hidden_state = sentiment_classifier(input_ids[i], attention_mask=attention_masks[i])[0]
        all_last_hidden_states.append(last_hidden_state)

# Concatenate the hidden states from each text
concatenated_last_hidden_states = torch.cat(all_last_hidden_states, dim=0)

print(f"last_hidden_state shape: {last_hidden_state.shape}")
print(f"concatenated_last_hidden_states shape: {concatenated_last_hidden_states.shape}")

# Check the size of the concatenated tensor
num_texts = input_ids.shape[0]
expected_rows = num_texts * max_num_aspects
actual_rows = concatenated_last_hidden_states.shape[0]
if actual_rows != expected_rows:
    print(f"Error: expected {expected_rows} rows but got {actual_rows} rows in concatenated_last_hidden_states")
else:
    print("Size check passed")

# Reshape the contextual embeddings to match the number of aspects
reshaped_embeddings = concatenated_last_hidden_states.view(num_texts,max_num_aspects, -1)

print(f"reshaped_embeddings shape: {reshaped_embeddings.shape}")



# Flatten the reshaped embeddings into a 2D matrix
flat_embeddings = reshaped_embeddings.reshape(num_texts, -1).numpy()


import numpy as np
from sklearn.model_selection import train_test_split

# Check the number of samples in flat_embeddings and labels
if len(flat_embeddings) > len(labels):
    n_samples = len(labels)
    indices = np.random.choice(len(flat_embeddings), n_samples, replace=False)
    flat_embeddings = flat_embeddings[indices]
elif len(labels) > len(flat_embeddings):
    n_samples = len(flat_embeddings)
    indices = np.random.choice(len(labels), n_samples, replace=False)
    labels = labels[indices]

# Split the flattened embeddings into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(flat_embeddings, labels, test_size=0.2, random_state=42)
```

```
# Create an XGBoost model
!pip install xgboost
import xgboost as xgb
xgb_model = xgb.XGBClassifier()
```

```
Requirement already satisfied: xgboost in c:\users\sivak\anaconda3\lib\site-packages (1.7.4)
Requirement already satisfied: scipy in c:\users\sivak\anaconda3\lib\site-packages (from xgboost) (1.10.0)
Requirement already satisfied: numpy in c:\users\sivak\anaconda3\lib\site-packages (from xgboost) (1.23.5)
```

```
import numpy as np
y_train = np.array([i[0] for i in y_train])
# Fit the model to the training data
xgb_model.fit(X_train, y_train)
```

```
▾                               XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, ...)
```

```
# Make predictions on the test data
y_pred = xgb_model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import MultiLabelBinarizer
import numpy as np

# Reshape y_pred into a 2D array
y_pred = np.reshape(y_pred, (-1, 1))

# Convert the labels into a binary array
mlb = MultiLabelBinarizer()
y_test_bin = mlb.fit_transform(y_test)
y_pred_bin = mlb.transform(y_pred)

# Compute the metrics using the binary arrays
print('Accuracy:', accuracy_score(y_test_bin, y_pred_bin))
print('Precision:', precision_score(y_test_bin, y_pred_bin, average='weighted'))
print('Recall:', recall_score(y_test_bin, y_pred_bin, average='weighted'))
print('F1-Score:', f1_score(y_test_bin, y_pred_bin, average='weighted'))
```

```
Accuracy: 0.5294117647058824
Precision: 0.5754475703324807
Recall: 0.6521739130434783
F1-Score: 0.6114130434782609
C:\Users\sivak\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision is ill-define
  _warn_prf(average, modifier, msg_start, len(result))
```