In [27]:
```
#Question 2
#Collect four documents on social matter from Wikipedia and produce their Td-Idf score.Expected.a. The four documentsb. Their score
```

In [21]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = [
            'Social media refers to new forms of media that involve interactive participation',
            'Public health is "the science and art of preventing disease, prolonging life and promoting health',
            'Child labour is the exploitation of children through any form of work that deprives them of their childhood,',
            'Black Lives Matter (BLM) is a decentralized political and social movement that seeks to highlight racism',
 ]

vectorizer = TfidfVectorizer()

# TD-IDF Matrix
X = vectorizer.fit_transform(corpus)

# extracting feature names
tfidf_tokens = vectorizer.get_feature_names_out()
```

In [22]:
```
import pandas as pd

result = pd.DataFrame(
    data=X.toarray(),
    index=["Doc1", "Doc2", "Doc3", "Doc4"],
    columns=tfidf_tokens
)

result
```

Out[22]:

	and	any	art	black	blm	child	childhood	children	decentralized	deprives	...	science	seeks	social	that	the	their	them	through
Doc1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.227047	0.183814	0.000000	0.000000	0.000000	0.000000
Doc2	0.395160	0.000000	0.250605	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.250605	0.000000	0.000000	0.000000	0.197580	0.000000	0.000000	0.000000
Doc3	0.000000	0.241803	0.000000	0.000000	0.000000	0.241803	0.241803	0.241803	0.000000	0.241803	...	0.000000	0.000000	0.000000	0.154340	0.190641	0.241803	0.241803	0.241803
Doc4	0.221412	0.000000	0.000000	0.280832	0.280832	0.000000	0.000000	0.000000	0.280832	0.000000	...	0.000000	0.280832	0.221412	0.179252	0.000000	0.000000	0.000000	0.000000

4 rows × 45 columns

In [23]:
```
# Creating the vectorizer
vectorizer = TfidfVectorizer(stop_words='english')

# Converting the text to TF-IDF matrix
X = vectorizer.fit_transform(corpus)

# Creating a DataFrame to display the results
tfidf_df = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out())
tfidf_df.index = ['Document {}'.format(i+1) for i in range(len(corpus))]
```

In [24]:
```
# Displaying the TF-IDF scores
print(tfidf_df)
```

```
                 art      black        blm     child  childhood  children  \
Document 1  0.000000  0.000000  0.000000  0.000000   0.000000  0.000000
Document 2  0.288675  0.000000  0.000000  0.000000   0.000000  0.000000
Document 3  0.000000  0.000000  0.000000  0.353553   0.353553  0.353553
Document 4  0.000000  0.306835  0.306835  0.000000   0.000000  0.000000

            decentralized  deprives   disease  exploitation  ...  preventing  \
Document 1       0.000000  0.000000  0.000000      0.000000  ...    0.000000
Document 2       0.000000  0.000000  0.288675      0.000000  ...    0.288675
Document 3       0.000000  0.353553  0.000000      0.353553  ...    0.000000
Document 4       0.306835  0.000000  0.000000      0.000000  ...    0.000000

            prolonging  promoting    public    racism    refers   science  \
Document 1    0.000000   0.000000  0.000000  0.000000  0.306835  0.000000
Document 2    0.288675   0.288675  0.288675  0.000000  0.000000  0.288675
Document 3    0.000000   0.000000  0.000000  0.000000  0.000000  0.000000
Document 4    0.000000   0.000000  0.000000  0.306835  0.000000  0.000000

                seeks    social      work
Document 1  0.000000  0.241912  0.000000
Document 2  0.000000  0.000000  0.000000
Document 3  0.000000  0.000000  0.353553
Document 4  0.306835  0.241912  0.000000

[4 rows x 35 columns]
```

In [25]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Defining the documents and the  labels
corpus = [
    'Social media refers to new forms of media that involve interactive participation',
    'Public health is "the science and art of preventing disease, prolonging life and promoting health',
    'Child labour is the exploitation of children through any form of work that deprives them of their childhood,',
    'Black Lives Matter (BLM) is a decentralized political and social movement that seeks to highlight racism',
]

labels = ['Technology', 'Health', 'Child Labor', 'Social Justice']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(corpus, labels, test_size=0.2, random_state=42)

# Create the TF-IDF vectorizer
vectorizer = TfidfVectorizer(stop_words='english')

# Transform the training data
X_train_tfidf = vectorizer.fit_transform(X_train)

# Train a text classification model (e.g., Multinomial Naive Bayes)
classifier = MultinomialNB()
classifier.fit(X_train_tfidf, y_train)

# Transform the test data
X_test_tfidf = vectorizer.transform(X_test)

# Make predictions
predictions = classifier.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
classification_report_output = classification_report(y_test, predictions)

print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:")
print(classification_report_output)
```

```
Accuracy: 0.00

Classification Report:
              precision    recall  f1-score   support

 Child Labor       0.00      0.00      0.00       0.0
      Health       0.00      0.00      0.00       1.0

    accuracy                           0.00       1.0
   macro avg       0.00      0.00      0.00       1.0
weighted avg       0.00      0.00      0.00       1.0
```

In [ ]:
```
The model failed to make any accurate predictions on the test set.
This could be due to a an imbalanced dataset and insufficient training data.
```

In [ ]:

In [ ]:
```
#How to classify text using Word2Vec
```

In [29]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')

# Reading the data
TicketData=pd.read_csv("C:/Users/hp/OneDrive/Desktop/supportTicketData.csv")

# Printing number of rows and columns
print(TicketData.shape)

# Printing sample rows
TicketData.head(10)
```

```
(19796, 2)
```

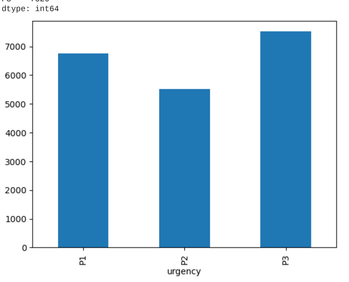
Out[29]:

	body	urgency
0	connection issues with assigned address hi fac...	P1
1	cannot access hi cannot access fallowing link ...	P2
2	re address shown valid dear colleagues remarke...	P1
3	sent tuesday critical alert following alert oc...	P2
4	code spelling mistake hello should discover fo...	P2
5	annual leave hello sent last week about previo...	P2
6	report working hello dear last two weeks have ...	P2
7	more access lost access please reset password ...	P1
8	open credentials required please assist instal...	P1
9	dear please ask our supplier for price quotati...	P2

In [30]:
```
# You can see there are 3 ticket types
print(TicketData.groupby('urgency').size())

# Plotting the bar chart
%matplotlib inline
TicketData.groupby('urgency').size().plot(kind='bar');
```

```
urgency
P1    6748
P2    5528
P3    7520
dtype: int64
```



In [31]:
```
# Count vectorization of text
from sklearn.feature_extraction.text import CountVectorizer

# Ticket Data
corpus = TicketData['body'].values

# Creating the vectorizer
vectorizer = CountVectorizer(stop_words='english')

# Converting the text to numeric data
X = vectorizer.fit_transform(corpus)

#print(vectorizer.get_feature_names())

# Preparing Data frame For machine learning
# Priority column acts as a target variable and other columns as predictors
CountVectorizedData=pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names())
CountVectorizedData['Priority']=TicketData['urgency']
print(CountVectorizedData.shape)
CountVectorizedData.head()
```

```
(19796, 9100)
```

Out[31]:

	ab	abandon	abandoned	abc	abeam	abilities	ability	able	abnormal	abnormally	...	zig	zip	zipped	zipper	zipping	zone	zones	zoom	zooming	Priority
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	P1
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	P2
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	P1
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	P2
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	P2

5 rows × 9100 columns

In [ ]: