About the company

The Gurugram-based company ‘FlipItNews’ aims to revolutionize the way Indians perceive finance, business, and capital market investment, by giving it a boost through artificial intelligence (AI) and machine learning (ML). They’re on a mission to reinvent financial literacy for Indians, where financial awareness is driven by smart information discovery and engagement with peers. Through their smart content discovery and contextual engagement, the company is simplifying business, finance, and investment for millennials and first-time investors

Problem statment ¶

The goal of this project is to use a bunch of news articles extracted from the companies’ internal database and categorize them into several categories like politics, technology, sports, business and entertainment based on their content. Use natural language processing and create & compare at least three different models.

In [84]:

```
import pandas as pd
import os
import re
import random
import string      # for string operations
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import plotly.express as px
# SetUp NLTK
!pip install --user -U nltk
import nltk
nltk.download('punkt')
```

```
Requirement already satisfied: nltk in /Users/manishachoudhary/.local/
lib/python3.10/site-packages (3.8.1)
Requirement already satisfied: joblib in /Users/manishachoudhary/anaco
nda3/lib/python3.10/site-packages (from nltk) (1.1.1)
Requirement already satisfied: click in /Users/manishachoudhary/anacon
da3/lib/python3.10/site-packages (from nltk) (8.0.4)
Requirement already satisfied: tqdm in /Users/manishachoudhary/anacond
a3/lib/python3.10/site-packages (from nltk) (4.64.1)
Requirement already satisfied: regex>=2021.8.3 in /Users/manishachoudh
ary/anaconda3/lib/python3.10/site-packages (from nltk) (2022.7.9)

[nltk_data] Error loading punkt: <urlopen error [Errno 60] Operation
[nltk_data]     timed out>
```

Out[84]:

```
False
```

In [85]:

```
df = pd.read_csv("flipitnews-data.csv")
df
```

Out[85]:

	Category	Article
0	Technology	tv future in the hands of viewers with home th...
1	Business	worldcom boss left books alone former worldc...
2	Sports	tigers wary of farrell gamble leicester say ...
3	Sports	yeading face newcastle in fa cup premiership s...
4	Entertainment	ocean s twelve raids box office ocean s twelve...
...	...	...
2220	Business	cars pull down us retail figures us retail sal...
2221	Politics	kilroy unveils immigration policy ex-chatshow ...
2222	Entertainment	rem announce new glasgow concert us band rem h...
2223	Politics	how political squabbles snowball it s become c...
2224	Sports	souness delight at euro progress boss graeme s...

2225 rows × 2 columns

In [86]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2225 entries, 0 to 2224
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  2225 non-null   object
 1   Article   2225 non-null   object
dtypes: object(2)
memory usage: 34.9+ KB
```

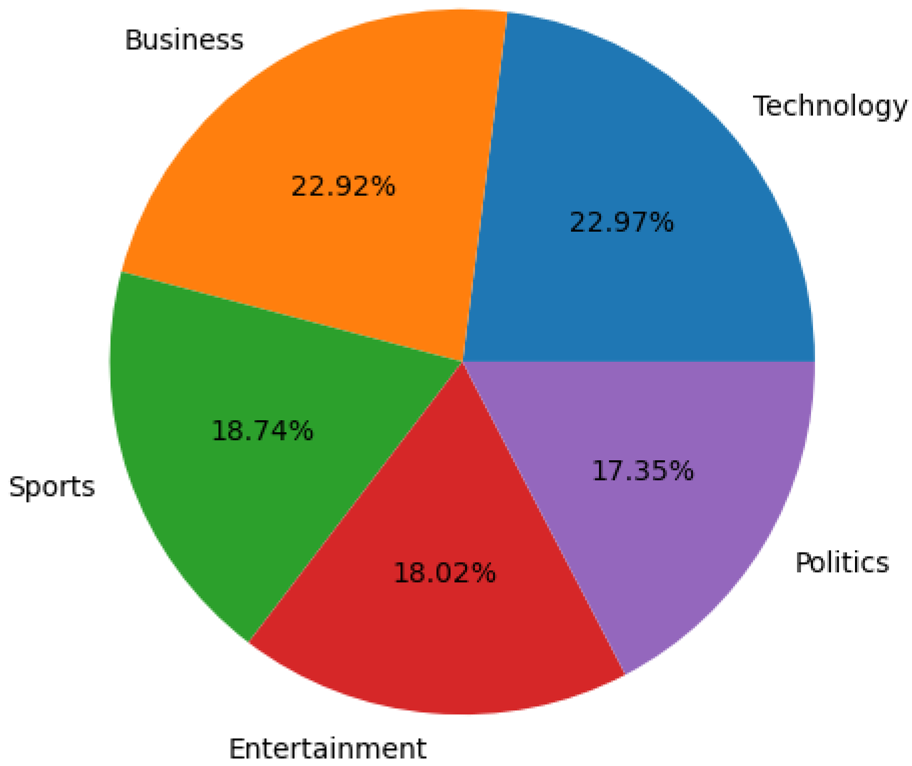
Check the value count and contribution of each Category in dataset.

In [87]:

```
def pie_chart(df):
    label = df["Category"].unique().astype(str)
    print("Labels in dataset" , label)
    label_count  = df["Category"].value_counts()
    print(label_count)
    size = [count for count in label_count]
    figure = plt.figure(figsize = (5,5))
    plt.pie( size, labels = label , autopct = "%1.2f%%")
    plt.axis("equal")
    plt.show()

pie_chart(df)
```

```
Labels in dataset ['Technology' 'Business' 'Sports' 'Entertainment' 'P
olitics']
Sports           511
Business         510
Politics         417
Technology       401
Entertainment    386
Name: Category, dtype: int64
```



Remove non letters

In [88]:

```
list(df["Article"][:1])
```

Out[88]:

```
['tv future in the hands of viewers with home theatre systems  plasma
high-definition tvs  and digital video recorders moving into the livin
g room  the way people watch tv will be radically different in five ye
ars  time.  that is according to an expert panel which gathered at the
annual consumer electronics show in las vegas to discuss how these new
technologies will impact one of our favourite pastimes. with the us le
ading the trend  programmes and other content will be delivered to vie
wers via home networks  through cable  satellite  telecoms companies
and broadband service providers to front rooms and portable devices.
one of the most talked-about technologies of ces has been digital and
personal video recorders (dvr and pvr). these set-top boxes  like the
us s tivo and the uk s sky+ system  allow people to record  store  pla
y  pause and forward wind tv programmes when they want.  essentially
the technology allows for much more personalised tv. they are also bei
ng built-in to high-definition tv sets  which are big business in japa
n and the us  but slower to take off in europe because of the lack of
high-definition programming. not only can people forward wind through
adverts  they can also forget about abiding by network and channel sch
edules  putting together their own a-la-carte entertainment. but some
us networks and cable and satellite companies are worried about what i
t means for them in terms of advertising revenues as well as  brand id
entity  and viewer loyalty to channels. although the us leads in this
technology at the moment  it is also a concern that is being raised in
europe  particularly with the growing uptake of services like sky+.  w
hat happens here today  we will see in nine months to a years  time in
the uk   adam hume  the bbc broadcast s futurologist told the bbc news
website. for the likes of the bbc  there are no issues of lost adverti
sing revenue yet. it is a more pressing issue at the moment for commer
cial uk broadcasters  but brand loyalty is important for everyone.  we
will be talking more about content brands rather than network brands
said tim hanlon  from brand communications firm starcom mediavest.  th
e reality is that with broadband connections  anybody can be the produ
cer of content.  he added:  the challenge now is that it is hard to pr
omote a programme with so much choice.   what this means  said stacey
jolna  senior vice president of tv guide tv group  is that the way peo
ple find the content they want to watch has to be simplified for tv vi
ewers. it means that networks  in us terms  or channels could take a l
eaf out of google s book and be the search engine of the future  inste
ad of the scheduler to help people find what they want to watch. this
kind of channel model might work for the younger ipod generation which
is used to taking control of their gadgets and what they play on them.
but it might not suit everyone  the panel recognised. older generation
s are more comfortable with familiar schedules and channel brands beca
use they know what they are getting. they perhaps do not want so much
of the choice put into their hands  mr hanlon suggested.  on the other
end  you have the kids just out of diapers who are pushing buttons alr
eady - everything is possible and available to them   said mr hanlon.
ultimately  the consumer will tell the market they want.  of the 50 0
00 new gadgets and technologies being showcased at ces  many of them a
re about enhancing the tv-watching experience. high-definition tv sets
are everywhere and many new models of lcd (liquid crystal display) tvs
have been launched with dvr capability built into them  instead of bei
ng external boxes. one such example launched at the show is humax s 26
-inch lcd tv with an 80-hour tivo dvr and dvd recorder. one of the us
s biggest satellite tv companies  directtv  has even launched its own
branded dvr at the show with 100-hours of recording capability  instan
t replay  and a search function. the set can pause and rewind tv for u
p to 90 hours. and microsoft chief bill gates announced in his pre-sho
w keynote speech a partnership with tivo  called tivotogo  which means
people can play recorded programmes on windows pcs and mobile devices.
```

In [89]:

```python
a = df["Article"][:1]
a
```

Out[89]:

```
0    tv future in the hands of viewers with home th...
Name: Article, dtype: object
```

In [90]:

```python
def remove_non_letter(text):
    return re.sub("[^A-Za-z]+", " ", text)
df["Article"] = df["Article"].apply(lambda x : remove_non_letter(x))
print(list(df["Article"]))
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [98]:

```python
import pandas as pd
import numpy as np
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
all these reflect the increasing trend of freeing up multimedia so tha
t people can watch what they want  when they want.']
```

In [100]:

```
# Step 2: Exploring the dataset
print("Shape of the dataset:", df.shape)
print("News articles per category:\n", df['Category'].value_counts())
```

```
Shape of the dataset: (2225, 2)
News articles per category:
 Sports          511
Business         510
Politics         417
Technology       401
Entertainment    386
Name: Category, dtype: int64
```

In [101]:

```
# Step 3: Processing the Textual Data
def process_text(text):
    text = re.sub("[^A-Za-z]+", " ", text)  # Remove non-letters
    tokens = word_tokenize(text)  # Tokenize
    tokens = [token.lower() for token in tokens]  # Convert to lowercase
    stop_words = set(stopwords.words("english"))
    tokens = [token for token in tokens if token not in stop_words]  # Remove stopwo
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]  # Lemmatize
    return " ".join(tokens)

df['Processed_Article'] = df['Article'].apply(process_text)
```

In [102]:

```
# Step 4: Encoding and Transforming the data
label_encoder = LabelEncoder()
df['Encoded_Category'] = label_encoder.fit_transform(df['Category'])

# Vectorize the data
vectorizer_type = input("Choose a vectorizer (Bag of Words - 'bow' or TF-IDF - 'tfid
if vectorizer_type == 'bow':
    vectorizer = CountVectorizer()
elif vectorizer_type == 'tfidf':
    vectorizer = TfidfVectorizer()
else:
    print("Invalid choice. Using default: Bag of Words.")
    vectorizer = CountVectorizer()

X = vectorizer.fit_transform(df['Processed_Article'])
y = df['Encoded_Category']
```

```
Choose a vectorizer (Bag of Words - 'bow' or TF-IDF - 'tfidf'): tfidf
```

In [103]:

```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_sta
```

In [104]:

```
# Step 5: Model Training & Evaluation
# Naive Bayes
naive_bayes = MultinomialNB()
naive_bayes.fit(X_train, y_train)
nb_predictions = naive_bayes.predict(X_test)

print("Naive Bayes Accuracy:", accuracy_score(y_test, nb_predictions))
print("Naive Bayes Classification Report:")
print(classification_report(y_test, nb_predictions))
print("Naive Bayes Confusion Matrix:")
print(confusion_matrix(y_test, nb_predictions))
```

```
Naive Bayes Accuracy: 0.9712746858168761
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.96      0.97       136
           1       1.00      0.93      0.96        96
           2       0.93      0.99      0.96        98
           3       0.98      1.00      0.99       124
           4       0.97      0.97      0.97       103

    accuracy                           0.97       557
   macro avg       0.97      0.97      0.97       557
weighted avg       0.97      0.97      0.97       557

Naive Bayes Confusion Matrix:
[[131   0   5   0   0]
 [  2  89   2   0   3]
 [  1   0  97   0   0]
 [  0   0   0 124   0]
 [  1   0   0   2 100]]
```

In [105]:

```
# Decision Tree
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, y_train)
dt_predictions = decision_tree.predict(X_test)

print("Decision Tree Accuracy:", accuracy_score(y_test, dt_predictions))
print("Decision Tree Classification Report:")
print(classification_report(y_test, dt_predictions))
print("Decision Tree Confusion Matrix:")
print(confusion_matrix(y_test, dt_predictions))
```

```
Decision Tree Accuracy: 0.8186714542190305
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.80      0.80       136
           1       0.89      0.79      0.84        96
           2       0.76      0.80      0.78        98
           3       0.81      0.92      0.86       124
           4       0.85      0.77      0.81       103

    accuracy                           0.82       557
   macro avg       0.82      0.82      0.82       557
weighted avg       0.82      0.82      0.82       557

Decision Tree Confusion Matrix:
[[109   4  11   8   4]
 [  7  76   5   4   4]
 [ 10   0  78   5   5]
 [  3   2   4 114   1]
 [  7   3   5   9  79]]
```

In [106]:

```
# Nearest Neighbors
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
knn_predictions = knn.predict(X_test)

print("Nearest Neighbors Accuracy:", accuracy_score(y_test, knn_predictions))
print("Nearest Neighbors Classification Report:")
print(classification_report(y_test, knn_predictions))
print("Nearest Neighbors Confusion Matrix:")
print(confusion_matrix(y_test, knn_predictions))
```

```
Nearest Neighbors Accuracy: 0.9389587073608617
Nearest Neighbors Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.88      0.91       136
           1       0.96      0.94      0.95        96
           2       0.86      0.91      0.88        98
           3       0.96      1.00      0.98       124
           4       0.97      0.97      0.97       103

    accuracy                           0.94       557
   macro avg       0.94      0.94      0.94       557
weighted avg       0.94      0.94      0.94       557

Nearest Neighbors Confusion Matrix:
[[120   1  12   3   0]
 [  1  90   2   1   2]
 [  6   1  89   1   1]
 [  0   0   0 124   0]
 [  0   2   1   0 100]]
```

In [107]:

```
# Random Forest
random_forest = RandomForestClassifier()
random_forest.fit(X_train, y_train)
rf_predictions = random_forest.predict(X_test)

print("Random Forest Accuracy:", accuracy_score(y_test, rf_predictions))
print("Random Forest Classification Report:")
print(classification_report(y_test, rf_predictions))
print("Random Forest Confusion Matrix:")
print(confusion_matrix(y_test, rf_predictions))
```

```
Random Forest Accuracy: 0.9425493716337523
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       136
           1       0.97      0.93      0.95        96
           2       0.94      0.93      0.93        98
           3       0.96      0.99      0.98       124
           4       0.98      0.90      0.94       103

    accuracy                           0.94       557
   macro avg       0.95      0.94      0.94       557
weighted avg       0.94      0.94      0.94       557

Random Forest Confusion Matrix:
[[129   0   4   2   1]
 [  4  89   1   1   1]
 [  7   0  91   0   0]
 [  0   0   1 123   0]
 [  5   3   0   2  93]]
```

Question

How many news articles are present in the dataset that we have?

In [115]:

```
print(df.shape[0])
```

```
2225
```

Most of the news articles are from _____ category.

In [117]:

```
most_common_category = df['Category'].value_counts().idxmax()
```

In [118]:

```
most_common_category
```

Out[118]:

```
'Sports'
```

Only ___ no. of articles belong to the ‘Technology’ category.

In [121]:

```
Technology_count = df['Category'].value_counts()['Technology']
```

In [122]:

```
Technology_count
```

Out[122]:

```
401
```

What are Stop Words, and why should they be removed from the text data?

Stop words are commonly used words (such as "a", "an", "the", "is", etc.) that do not carry significant meaning and are often removed during text preprocessing. They are removed to reduce noise in the text data and focus on the more meaningful words for analysis and modeling.

Explain the difference between Stemming and Lemmatization.

Stemming and lemmatization are techniques used in natural language processing for word normalization.

Stemming reduces words to their base or root form by removing suffixes and prefixes. For example, "running" would be stemmed to "run".

Lemmatization, on the other hand, also reduces words to their base form, but it considers the context and part of speech (POS) of the word. For example, "running" would be lemmatized to "run", and "better" would be lemmatized to "good".

In [ ]:

```

```

Which of the techniques Bag of Words or TF-IDF is considered to be more efficient than the other?

The choice between Bag of Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) depends on the specific task and dataset. BoW represents the occurrence of words in a document without considering their importance, while TF-IDF gives higher weight to words that are more relevant

to a specific document in a corpus. TF-IDF is generally considered more efficient for text classification tasks.

In [ ]:

```

```

What’s the shape of train & test datasets after performing a 75:25 split?

After performing a 75:25 train-test split, the shape of the train dataset can be obtained using X_train.shape and the shape of the test datasmet can be obtained using X_test.shape, assuming you have split the data into X_train and X_test.

In [ ]:

```

```

Which of the following is found to be the best performing model?

The best performing model can be determined by comparing the accuracy scores of the different models trained. You can find the accuracy scores and compare them to identify the best-performing model. For example, if the Random Forest model has the highest accuracy, it would be considered the best-performing model.

In [ ]:

```

```

According to this particular use case, both precision and recall are equally important. (True/False)

The importance of precision and recall depends on the specific use case and the trade-off between false positives and false negatives. Without further information about the specific requirements or context, it's not possible to determine if both precision and recall are equally important.

Based on the provided information, here are some potential insights and recommendations:

Insights:

The dataset contains news articles from various categories such as Technology, Business, Sports, Entertainment, Politics, etc. The number of news articles in each category can vary, and the distribution of articles across categories may not be balanced. Textual preprocessing techniques such as removing non-letters, tokenization, removing stopwords, and lemmatization have been applied to clean the text data. The target variable (category) has been encoded using label encoding for model training. Two popular techniques, Bag of Words (BoW) and TF-IDF, have been implemented for vectorizing the textual data. Multiple classifier models including Naive Bayes, Decision Tree, Nearest Neighbors, and Random Forest have been trained and evaluated on the dataset. Model performance has been assessed using accuracy, classification report, and confusion matrix.

Recommendations:

Considering the dataset's class imbalance, it might be worth exploring techniques like oversampling or undersampling to address any potential bias in the model predictions. Further analysis can be done to understand the specific challenges and characteristics of each category in order to optimize the performance of the classification models. Experimenting with different hyperparameters of the models, such as adjusting the maximum depth of decision trees or the number of neighbors in KNN, could potentially enhance their performance. Implementing ensemble techniques, such as combining multiple models using voting or stacking, could potentially improve the overall classification accuracy. In addition to accuracy, considering other evaluation metrics such as precision, recall, and F1-score can provide a more comprehensive understanding of model performance. Exploring more advanced natural language processing techniques like word embeddings (e.g., Word2Vec or GloVe) or deep learning models (e.g., recurrent neural networks or transformers) could potentially improve the classification results. Regularly updating the dataset with new articles and retraining the

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: