


> Importing the necessary Libraries

[] ↳ 1 cell hidden

✓ Import Dataset

```
1 from google.colab import files
2 uploaded = files.upload()
```

 Choose Files | bbc-news-data.csv

- **bbc-news-data.csv**(text/csv) - 5080260 bytes, last modified: 7/31/2020 - 100% done

Saving bbc-news-data.csv to bbc-news-data (1).csv

```
1 import pandas as pd                #Imports the pandas library as pd
2 import numpy as np                 #Imports the numpy library for numerical computing
3
4 data_file="bbc-news-data.csv"
5
6 data = pd.read_csv(data_file,sep='\t')    ## Load the CSV file into a DataFrame
```

```
1 data.head(10)
```



index	category	filename	title	content
0	business	001.txt	Ad sales boost Time Warner profit	Quarterly profits at US media giant TimeWarner jumped 76% to \$1.13bn (£600m) for the three months to December, from \$639m year-earlier. The firm, which is now one of the biggest investors in Google, benefited from sales of high-speed internet connections and higher advert sales. TimeWarner said fourth quarter sales rose 2% to \$11.1bn from \$10.9bn. Its profits were buoyed by one-off gains which offset a profit dip at Warner Bros, and less users for AOL. Time Warner said on Friday that it now owns 8% of search-engine Google. But its own internet business, AOL, had has mixed fortunes. It lost 464,000 subscribers in the fourth quarter profits were lower than in the preceding three quarters. However, the company said AOL's underlying profit before exceptional items rose 8% on the back of stronger internet advertising revenues. It hopes to increase subscribers by offering the online service free to TimeWarner internet customers and will try to sign up AOL's existing customers for high-speed broadband. TimeWarner also has to restate 2000 and 2003 results following a probe by the US Securities Exchange Commission (SEC), which is close to concluding. Time Warner's fourth quarter profits were slightly better than analysts' expectations. But its film division saw profits slump 27% to \$284m, helped by box-office flops Alexander and Catwoman, a sharp contrast to year-earlier, when the third and final film in the Lord of the Rings trilogy boosted results. For the full-year, TimeWarner posted a profit of \$3.36bn, up 27% from its 2003 performance, while revenues grew 6.4% to \$42.09bn. "Our financial performance was strong, meeting or exceeding all of our full-year objectives and greatly enhancing our flexibility," chairman and chief executive Richard Parsons said. For 2005, TimeWarner is projecting operating earnings growth of around 5%, and also expects higher revenue and wider profit margins. TimeWarner is to restate its accounts as part of efforts to resolve an inquiry into AOL by US market regulators. It has already offered to pay \$300m to settle charges, in a deal that is under review by the SEC. The company said it was unable to estimate the amount it needed to set aside for legal reserves, which it previously set at \$500m. It intends to adjust the way it accounts for a deal with German music publisher Bertelsmann's purchase of a stake in AOL Europe, which it had reported as advertising revenue. It will now book the sale of its stake in AOL Europe as a loss on the value of that stake.
1	business	002.txt	Dollar gains on Greenspan speech	The dollar has hit its highest level against the euro in almost three months after the Federal Reserve head said the US trade deficit is set to stabilise. And Alan Greenspan highlighted the US government's willingness to curb spending and rising household savings as factors which may help to reduce it. In late trading in New York, the dollar reached \$1.2871 against the euro, from \$1.2974 on Thursday. Market concerns about the deficit has hit the greenback in recent months. On Friday, Federal Reserve chairman Mr Greenspan's speech in London ahead of the meeting of G7 finance ministers sent the dollar higher after it had earlier tumbled on the back of worse-than-expected US jobs data. "I think the chairman's taking a much more sanguine view on the current account deficit than he's taken for some time," said Robert Sinche, head of currency strategy at Bank of America in New York. "He's taking a longer-term view, laying out a set of conditions under which the current account deficit can improve this year and next." Worries about the deficit concerns about China do, however, remain. China's currency remains pegged to the dollar and the US currency's sharp falls in recent months have therefore made Chinese export prices highly competitive. But calls for a shift in Beijing's policy have fallen on deaf ears, despite recent comments in a major Chinese newspaper that the "time is ripe" for a loosening of the peg. The G7 meeting is thought unlikely to produce any meaningful movement in Chinese policy. In the meantime, the US Federal Reserve's decision on 2 February to boost interest rates by a quarter of a point - the sixth such move in as many months - has opened up a differential with European rates. The half-point window, some believe, could be enough to keep US assets looking more attractive, and could help prop up the dollar. The recent falls have partly been the result of big budget deficits, as well as the US's yawning current account gap, both of which need to be funded by the buying of US bonds and assets by foreign firms and governments. The White House will announce its budget on Monday, and many commentators believe the deficit will remain at close to half a trillion dollars.
2	business	003.txt	Yukos unit buyer faces loan claim	The owners of embattled Russian oil giant Yukos are to ask the buyer of its former production unit to pay back a \$900m (£479m) loan. State-owned Rosneft bought the Yugansk unit for \$9.3bn in a sale forced by Russia to part settle a \$27.5bn tax claim against Yukos. Yukos' owner Menatep Group says it will ask Rosneft to repay a loan that Yukos had secured on its assets. Rosneft already faces a similar \$540m repayment demand from foreign banks. Legal experts said Rosneft's purchase of Yugansk would include such obligations. "The pledged assets are with Rosneft, so it will have to pay real money to the creditors to avoid seizure of Yugansk assets," said Moscow-based US lawyer Jamie Firestone, who is not connected to the case. Menatep Group's managing director Tim Osborne told the Reuters news agency: "If they default, we will fight them where the rule of law exists under the international arbitration clauses of the credit." Rosneft officials were unavailable for comment. But the company has said it intends to take action against Menatep to recover some of the tax claims and debts owed by Yugansk. Yukos had filed for bankruptcy protection in a US court in an attempt to prevent the forced sale of its main production arm. The sale went ahead in December and Yugansk was sold to a little-known shell company which in turn was bought by Rosneft. Yukos claims its downfall was punishment for the political ambitions of its founder Mikhail Khodorkovsky and has vowed to sue any participant in the sale.
3	business	004.txt	High fuel prices hit BA's profits	British Airways has blamed high fuel prices for a 40% drop in profits. Reporting its results for the three months to 31 December 2004, the airline made a pre-tax profit of £75m (\$141m) compared with £125m a year earlier. Rod Eddington, BA's chief executive, said the results were "respectable" in a third quarter when fuel costs rose by £106m or 47.3%. BA's profits were still better than market expectation of £59m, and it expects a rise in full-year revenues. To help offset the increased price of aviation fuel, BA last year introduced a fuel surcharge for passengers. In October, it increased this from £6 to £10 one-way for all long-haul flights, while the short-haul surcharge was raised from £2.50 to £4 a leg. Yet aviation analyst Mike Powell of Dresdner Kleinwort Wasserstein says BA's estimated annual surcharge revenues - £160m - will still be way short of its additional fuel costs - a predicted extra £250m. Turnover for the quarter was up 4.3% to £1.97bn, further benefiting from a rise in cargo revenue. Looking ahead to its full year results to March 2005, BA warned that yields - average revenues per passenger - were expected to decline as it continues to lower prices in the face of competition from low-cost carriers. However, it said sales would be better than previously forecast. "For the year to March 2005, the total revenue outlook is slightly better than previous guidance with a 3% to 3.5% improvement anticipated," BA chairman Martin Broughton said. BA had previously forecast a 2% to 3% rise in full-year revenue. It also reported on Friday that passenger numbers rose 8.1% in January. Aviation analyst Nick Van den Brul of BNP Paribas described BA's latest quarterly results as "pretty modest". "It is quite good on the revenue side and it shows the impact of fuel surcharges and a positive cargo development, however, operating margins down and cost impact of fuel are very strong," he said. Since the 11 September 2001 attacks in the United States, BA has cut 13,000 jobs as part of a major cost-cutting drive. "Our focus remains on reducing controllable costs and debt whilst continuing to invest in our products," Mr Eddington said. "For example, we have taken delivery of six Airbus A321 aircraft and next month we will start further improvements to our Club World flat beds." BA's shares closed up four pence at 274.5 pence.
4	business	005.txt	Pernod takeover talk lifts Domecq	Shares in UK drinks and food firm Allied Domecq have risen on speculation that it could be the target of a takeover by France's Pernod Ricard. Reports in the Wall Street Journal and the Financial Times suggested that the French spirits firm is considering a bid, but has yet to contact its target. Allied Domecq shares in London rose 4% by 1200 GMT, while Pernod shares in Paris slipped 1.2%. Pernod said it was seeking acquisitions but refused to comment on specifics. Pernod's last major purchase was a third of US giant Seagram in 2000, the move which propelled it into the global top three of drinks firms. The other two-thirds of Seagram was bought by market leader Diageo. In terms of market value, Pernod - at 7.5bn euros (\$9.7bn) - is about 9% smaller than Allied Domecq, which has a capitalisation of £5.7bn (\$10.7bn; 8.2bn euros). Last year Pernod tried to buy Glenmorangie, one of Scotland's premier whisky firms, but lost out to luxury goods firm LVMH. Pernod is home to brands including Chivas Regal Scotch whisky, Havana Club rum and Jacob's Creek wine. Allied Domecq's big names include Malibu rum, Courvoisier brandy, Stolichnaya vodka and Ballantine's whisky - as well as snack food chains such as Dunkin' Donuts and Baskin-Robbins ice cream. The WSJ said that the two were ripe for consolidation, having each dealt with problematic parts of their portfolio. Pernod has reduced the debt it took on to fund the Seagram purchase to just 1.8bn euros, while Allied has improved the performance of its fast-food chains.
5	business	006.txt	Japan narrowly escapes recession	Japan's economy teetered on the brink of a technical recession in the three months to September, figures show. Revised figures indicated growth of just 0.1% - and a similar-sized contraction in the previous quarter. On an annual basis, the data suggests annual growth of just 0.2%, suggesting a much more hesitant recovery than had previously been thought. A common technical definition of a recession is two successive quarters of negative growth. The government was keen to play down the worrying implications of the data. "I maintain the view that Japan's economy remains in a minor adjustment phase in an upward climb, and we will monitor developments carefully," said economy minister Heizo Takenaka. But in the face of the strengthening yen making exports less competitive and indications of weakening economic conditions ahead, observers were less sanguine. "It's painting a picture of a recovery... much patchier than previously thought," said Paul Sheard, economist at Lehman Brothers in Tokyo. Improvements in the job market apparently have yet to feed through to domestic demand, with private consumption up just 0.2% in the third quarter.
6	business	007.txt	Jobs growth still slow in the US	The US created fewer jobs than expected in January, but a fall in jobseekers pushed the unemployment rate to its lowest level in three years. According to Labor Department figures, US firms added only 146,000 jobs in January. The gain in non-farm payrolls was below market expectations of 190,000 new jobs. Nevertheless it was enough to push down the unemployment rate to 5.2%, its lowest level since September 2001. The job gains mean that President Bush can celebrate - albeit by a very fine margin - a net growth in jobs in the US economy in his first term in office. He presided over a net fall in jobs up to last November's Presidential election - the first President to do so since Herbert Hoover. As a result, job creation became a key issue in last year's election. However, when adding December and January's figures, the administration's first term jobs record ended in positive territory. The Labor Department also said it had revised down the jobs gains in December 2004, from 157,000 to 133,000. Analysts said the growth in new jobs was not as strong as could be expected given the favourable economic conditions. "It suggests that employment is continuing to expand at a moderate pace," said Rick Egelton, deputy chief economist at BMO Financial Group. "We are not getting the boost to employment that we would have got given the low value of the dollar and the still relatively low interest rate

Next steps:

Generate code with data



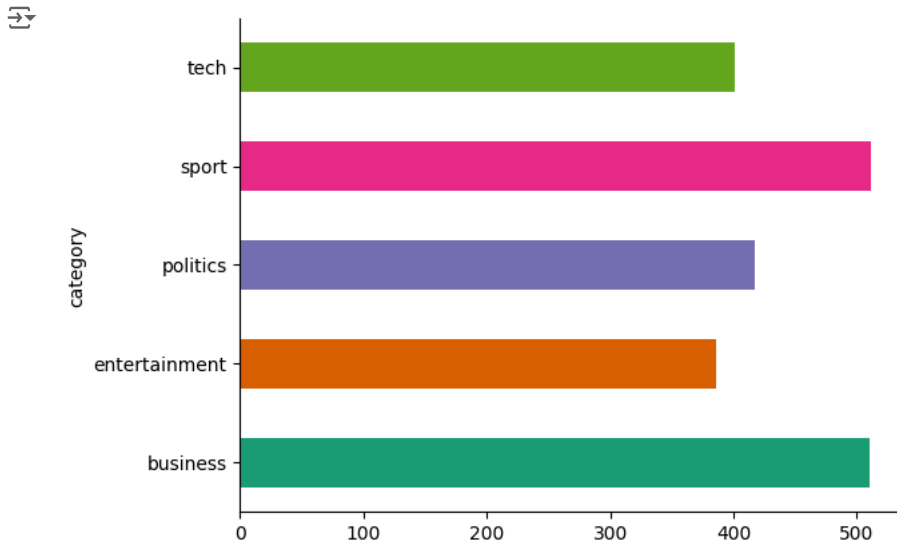
View recommended plots

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2225 entries, 0 to 2224
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   category    2225 non-null   object  
 1   filename    2225 non-null   object  
 2   title       2225 non-null   object  
 3   content     2225 non-null   object  
dtypes: object(4)
memory usage: 69.7+ KB
```

category

```
1
2
3 # @title category
4
5 from matplotlib import pyplot as plt
6 import seaborn as sns
7 data.groupby('category').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
8 plt.gca().spines[['top', 'right',]].set_visible(False)
```



Ydata Profiling

```
1 from ydata_profiling import ProfileReport
2 # Generate the data profiling report
3 ydatareport = ProfileReport(data)
4 #Display report
5 ydatareport
```



Summarize dataset: 100% 13/13 [00:02<00:00, 4.73it/s, Completed]

Generate report structure: 100% 1/1 [00:04<00:00, 4.97s/it]

Render HTML: 100% 1/1 [00:00<00:00, 2.17it/s]

Total size in memory	32.1 MB
Average record size in memory	32.1 B

Reproduction

Analysis started	2024-07-22 00:09:02.674365
Analysis finished	2024-07-22 00:09:04.812004
Duration	2.14 seconds
Software version	ydata-profiling vv4.9.0 (https://github.com/ydataai/ydata-profiling)
Download configuration	config.json (data:text/plain;charset=utf-8,%7B%22title%22%3A%20%22Pandas%20Profiling%20Report%22%2C%20%22dataset%22%3A%20%7B%22description%22%3A%20%22%22%22%7D%7D)

Variables

Select Columns ▾

category
Categorical

Distinct	5
Distinct (%)	0.2%
Missing	0
Missing (%)	0.0%

Text cleaning/Stop Words Removal

Removing irrelevant characters, punctuation, and extra spaces reduces noise in the data, allowing the model to focus on meaningful information. Converting text to lowercase ensures uniformity, so the model treats "The" and "the" as the same word. Removing stop words (common words like "the", "is", "in") reduces the number of features, focusing the model on more informative words. Clean data can lead to faster and more efficient training because the model has fewer irrelevant features to consider. Reducing feature dimensionality (e.g., through stemming and stop word removal) can lead to better generalization and less overfitting. Lemmatization reduces words to their base or root form. For example, "running" and "ran" both become "run". This normalization helps the model to better understand the context and meaning of words, leading to more accurate predictions. By reducing different forms of a word to a single form, lemmatization reduces the number of unique words (features). This helps the model to generalize better and avoid overfitting.

Text Cleaning without stemming/stop word removal

```
1 #import re                                #Import module for regular expressions
2
3 # Text cleaning function
4 #def clean_text(text):
5 #     text = re.sub(r'\s+', ' ', text) # to remove extra spaces, newlines, and tabs.
6 #     text = re.sub(r'\W', ' ', text)  # to replace all non-word characters with a space.
7 #     return text.strip().lower()      # to remove any leading and trailing whitespace from the text and Converts the text to lowercase.
```

Text Cleaning using stemming/stop word removal

```

1 # Text cleaning function
2 def clean_text(text):
3     text = re.sub(r'\W', ' ', text)
4     text = text.lower()
5     text = re.sub(r'\s+', ' ', text)
6     stop_words = set(stopwords.words('english'))
7     lemmatizer = WordNetLemmatizer()
8     #text = ' '.join([word for word in text.split() if word not in stop_words])
9     text = ' '.join([lemmatizer.lemmatize(word) for word in text.split() if word not in stop_words])
10    return text
11
12
13 # Clean the content
14 data['cleaned_content'] = data['content'].apply(clean_text)

```

✓ **Balancing Dataset**

In an imbalanced dataset, models can become biased towards the majority class. This means that the model will predict the majority class more often simply because there are more examples of it, which can result in poor performance on the minority class. So we balanced the dataset by reducing the number of examples in the majority class to match the minority class.

Key Research Question: • How does balancing of dataset affect the performance?

We have used the random sampling and stratified sampling and found that stratified sampling has improved the performance of three classifiers.

```

1 # Balance the dataset
2 min_samples = data['category'].value_counts().min()
3 balanced_data = data.groupby('category').apply(lambda x: x.sample(min_samples)).reset_index(drop=True)

```


✓ **TFIDF Vectorization**

Max Features in Content(text) Column Unigram

```

1 # Extract the content column
2 content = data['content']
3
4 # Use CountVectorizer to count unique terms
5 vectorizer = CountVectorizer(stop_words='english')
6 X_counts = vectorizer.fit_transform(content)
7
8 # Get the number of unique terms
9 num_unique_terms = len(vectorizer.get_feature_names_out())
10
11 print(f'Number of unique terms (Unigrams): {num_unique_terms}')

```

 Number of unique terms (Unigrams): 28980

Key research question:

• Is TF-idf effective for the features extraction from news articles?

News articles typically have a large vocabulary with many unique words. TF-IDF can handle this by giving more importance to unique and meaningful words and less to common words. News articles usually have sparse data, meaning most words appear only in a few documents. The total number of unique words are 28980. So we experimented with choosing the optimal number of max features and find that it was giving the best performance at max feature=5000. The more number of feature will use more resources.

```

1 # Vectorization
2 tfidf_vectorizer = TfidfVectorizer(max_features=5000)
3 X = tfidf_vectorizer.fit_transform(balanced_data['cleaned_content'])
4 y = balanced_data['category']

```

✓ **Splitting data into Train-test subset**

Stratified sampling is used so that each class (or stratum) is represented proportionally in both the training and test sets. This helps with imbalanced datasets because it maintains the same class distribution across splits, which helps in providing a more reliable estimate of model performance.

```

1 from sklearn.model_selection import StratifiedShuffleSplit, train_test_split
2 # Splitting the balanced data
3 sss = StratifiedShuffleSplit(n_splits=1, test_size=0.25, random_state=42)
4 for train_index, test_index in sss.split(X, y):
5     X_train, X_test = X[train_index], X[test_index]
6     y_train, y_test = y[train_index], y[test_index]
7
8 # Check the distribution of the categories in the test set
9 print("Category distribution in test set:")
10 print(y_test.value_counts())

```

```

Category distribution in test set:
category
politics      97
business      97
sport          97
tech          96
entertainment 96
Name: count, dtype: int64

```

X: The feature matrix (TF-IDF transformed text data). y: The labels (news categories). test_size=0.25: 25% of the data is reserved for testing, and 75% for training. random_state=42: A seed value for random number generation to ensure reproducibility. Using the same seed value will always produce the same train-test split.

Model Training and Evaluation

Below are the four functions for training and evaluation, confusion matrix, plotting confusion matrix and for cross-validation.

Cross-validation provides a more reliable estimate of a model's performance by averaging the results from multiple train-test splits. This reduces the likelihood of overestimating or underestimating the model's true performance. By using multiple folds, cross-validation reduces the variance associated with a single train-test split, leading to a more stable and reliable performance estimate. We used the 10 fold cross validation for all three algorithms. Ten-fold cross-validation is recommended as a good compromise between computational cost and accuracy estimation." (Kohavi, 1995)

```

1 from sklearn.metrics import accuracy_score, classification_report, precision_score, recall_score, confusion_matrix
2 import seaborn as sns
3
4 def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
5     model.fit(X_train, y_train)
6     y_pred = model.predict(X_test)
7     accuracy = accuracy_score(y_test, y_pred)
8     report = classification_report(y_test, y_pred, target_names=y_test.unique())
9     cm = confusion_matrix(y_test, y_pred, labels=y_test.unique())
10    return accuracy, report, cm, y_pred
11
12 # Calculate and print TP, FP, FN, TN
13 def calculate_metrics(cm):
14     for i in range(len(cm)):
15         tp = cm[i, i]
16         fp = cm[:, i].sum() - tp
17         fn = cm[i, :].sum() - tp
18         tn = cm.sum() - (tp + fp + fn)
19         print(f'Class {i}: TP: {tp}, FP: {fp}, TN: {tn}, FN: {fn}')
20
21 def plot_confusion_matrix(cm, classes):
22     plt.figure(figsize=(10, 7))
23     sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=classes, yticklabels=classes)
24     plt.ylabel('True label')
25     plt.xlabel('Predicted label')
26     plt.title('Confusion Matrix')
27     plt.show()
28
29 # Cross-validation
30 def cross_validate_model(model, X, y, cv=10):
31     scores = cross_val_score(model, X, y, cv=cv)
32     print(f"Cross-Validation Scores: {scores}")
33     print(f"Mean Accuracy: {scores.mean()}")
34     print(f"Standard Deviation: {scores.std()}")

```

Naive Bayes

Naive bayes Model training: Naive Bayes classifiers, particularly the Multinomial Naive Bayes, are well-suited for handling sparse data, such as text data represented as TF-IDF or count vectors.

```

1 nb_model = MultinomialNB()
2 nb_accuracy, nb_report, nb_cm, nb_y_pred = train_and_evaluate_model(
3     nb_model, X_train, y_train, X_test, y_test)

```

Evaluation_nb

```
1 # Print results
2 print("Naive Bayes Classifier:")
3 print("Accuracy:", nb_accuracy)
4 print("Classification Report:\n", nb_report)
```

Naive Bayes Classifier:
Accuracy: 0.9710144927536232
Classification Report:

	precision	recall	f1-score	support
tech	0.95	0.95	0.95	97
politics	0.99	0.96	0.97	96
business	0.94	0.99	0.96	97
sport	1.00	0.99	0.99	97
entertainment	0.98	0.97	0.97	96
accuracy			0.97	483
macro avg	0.97	0.97	0.97	483
weighted avg	0.97	0.97	0.97	483

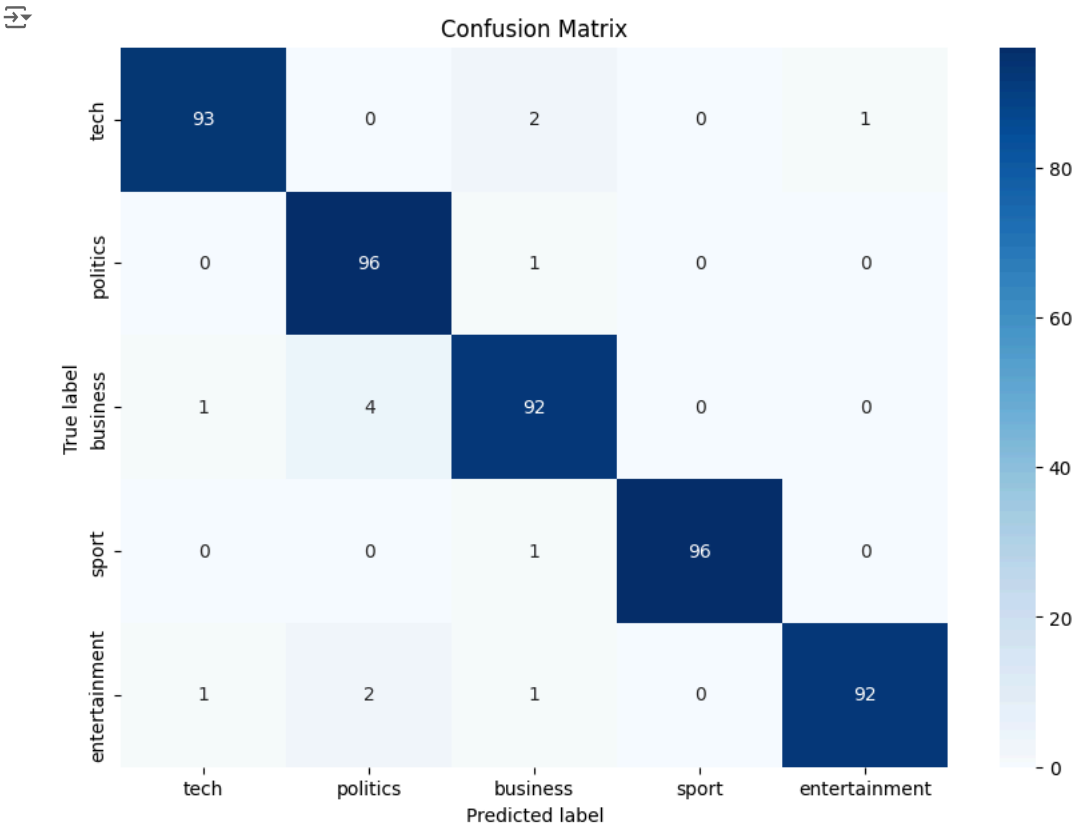
```
1
2 print("\nMetrics per Class:")
3 calculate_metrics(nb_cm)
```

Metrics per Class:

Class 0:	TP: 93, FP: 2, TN: 385, FN: 3
Class 1:	TP: 96, FP: 6, TN: 380, FN: 1
Class 2:	TP: 92, FP: 5, TN: 381, FN: 5
Class 3:	TP: 96, FP: 0, TN: 386, FN: 1
Class 4:	TP: 92, FP: 1, TN: 386, FN: 4

The above confusion matrix is for five news categories. Class 1 is tech and it shows that out of total 483 samples used for testing. It predicted 93 out of 97 tech texts correctly, and gave 381 True negatives.

```
1 # Plot the confusion matrix
2 plot_confusion_matrix(nb_cm, classes=y_test.unique())
```



Cross Validation_nb

We used the 10 fold cross validation for all three algorithms. Ten-fold cross-validation is recommended as a good compromise between computational cost and accuracy estimation." (Kohavi, 1995)

```
1 print("Naive Bayes Classifier (Cross-Validation):")
2 cross_validate_model(nb_model, X, y)
```



```
Naive Bayes Classifier (Cross-Validation):
Cross-Validation Scores: [0.95336788 0.96373057 0.96373057 0.97409326 0.97927461 0.98963731
0.97927461 0.98963731 0.97927461 0.97927461]
Mean Accuracy: 0.9751295336787564
Standard Deviation: 0.01106432979485111
```

Decision Tree

Decision trees do not handle sparse data as efficiently as models like Naive Bayes or linear models. Sparse matrices often contain many zero entries, which may not provide useful information for splits. So we desparse the sparse matrix to a dense format for decision tree implementation and find that performance was improved.

```
1 X_train_dense = X_train.toarray()
2 X_test_dense = X_test.toarray()
```

Decsion tree model training

```
1 dt_model = DecisionTreeClassifier(random_state=42)
2 dt_accuracy, dt_report, dt_cm, dt_y_pred = train_and_evaluate_model(
3     dt_model, X_train, y_train, X_test, y_test)
```

Evaluation_dt

```
1 print("Decision Tree Classifier:")
2 print("Accuracy:", dt_accuracy)
3 print("Classification Report:\n", dt_report)
```



```
Decision Tree Classifier:
Accuracy: 0.8385093167701864
Classification Report:
              precision    recall  f1-score   support

   tech          0.84        0.79        0.81         97
  politics          0.80        0.75        0.77         96
  business          0.81        0.81        0.81         97
    sport          0.90        0.93        0.91         97
entertainment          0.84        0.91        0.87         96

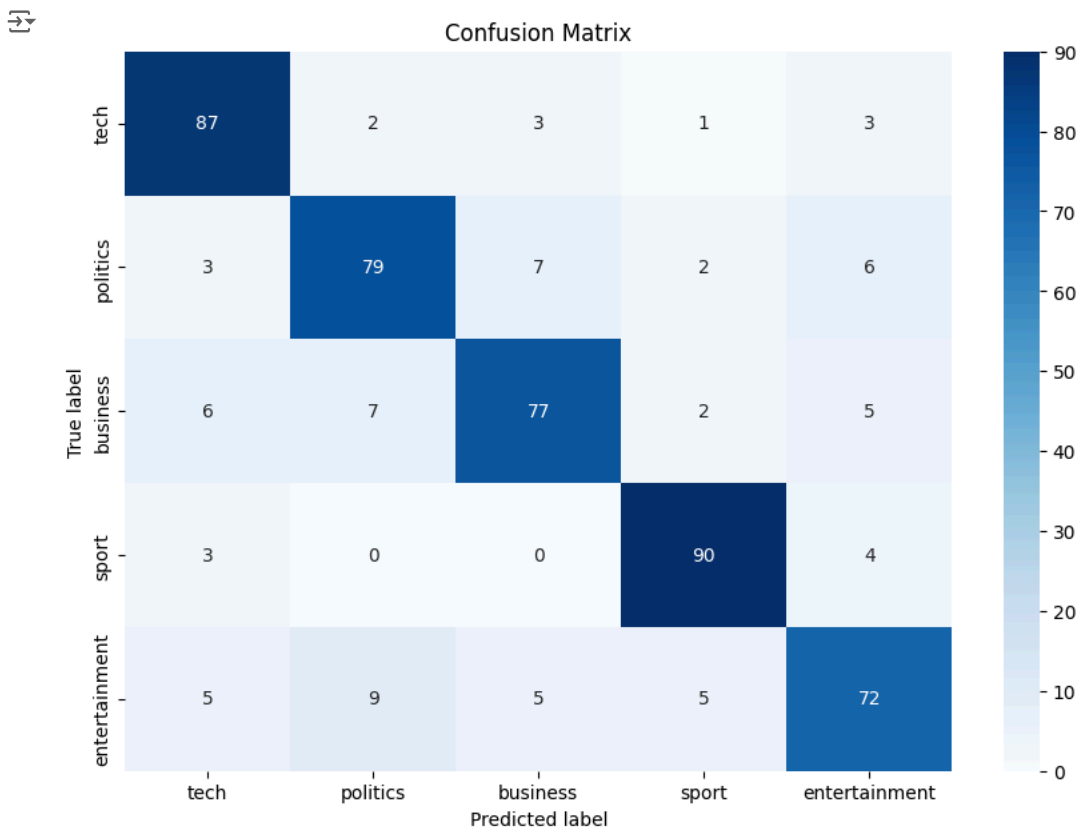
   accuracy          0.84          0.84        0.84        483
  macro avg          0.84        0.84        0.84        483
weighted avg          0.84        0.84        0.84        483
```

```
1 print("\nMetrics per Class:")
2 calculate_metrics(dt_cm)
```



```
Metrics per Class:
Class 0: TP: 87, FP: 17, TN: 370, FN: 9
Class 1: TP: 79, FP: 18, TN: 368, FN: 18
Class 2: TP: 77, FP: 15, TN: 371, FN: 20
Class 3: TP: 90, FP: 10, TN: 376, FN: 7
Class 4: TP: 72, FP: 18, TN: 369, FN: 24
```

```
1 # Plot the confusion matrix
2 plot_confusion_matrix(dt_cm, classes=y_test.unique())
```

Cross Validation_dt

```
1 print("Decision Tree Classifier (Cross-Validation):")
2 cross_validate_model(dt_model, X, y)
```

Decision Tree Classifier (Cross-Validation):
Cross-Validation Scores: [0.85492228 0.81865285 0.77720207 0.8134715 0.80829016 0.89119171
0.83419689 0.83419689 0.86528497 0.8238342]
Mean Accuracy: 0.8321243523316062
Standard Deviation: 0.03049521029961069

Random Forest

```
1 # Random Forest
2 rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
3 rf_accuracy, rf_report, rf_cm, rf_y_pred = train_and_evaluate_model(
4     rf_model, X_train, y_train, X_test, y_test)
```

Evaluation-rf

```
1 print("Random Forest Classifier:")
2 print("Accuracy:", rf_accuracy)
3 print("Classification Report:\n", rf_report)
```

```
Random Forest Classifier:
Accuracy: 0.9523809523809523
Classification Report:
              precision    recall  f1-score   support

   tech       0.90      0.97      0.94         97
  politics    0.96      0.94      0.95         96
  business    0.96      0.92      0.94         97
    sport     0.97      0.98      0.97         97
entertainment 0.98      0.96      0.97         96

   accuracy          0.95
  macro avg          0.95
 weighted avg          0.95
```

```
1 print("\nMetrics per Class:")
2 calculate_metrics(rf_cm)
```

```
Metrics per Class:
Class 0: TP: 92, FP: 2, TN: 385, FN: 4
Class 1: TP: 89, FP: 4, TN: 382, FN: 8
Class 2: TP: 94, FP: 10, TN: 376, FN: 3
Class 3: TP: 95, FP: 3, TN: 383, FN: 2
Class 4: TP: 90, FP: 4, TN: 383, FN: 6
```

```
1 # Plot the confusion matrix
2 plot_confusion_matrix(rf_cm, classes=y_test.unique())
```

↗

Confusion Matrix

