UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Behavior Analysis Technologies

Session 5

Text Classification

Applied Data Science

2024/2025

Text Classification

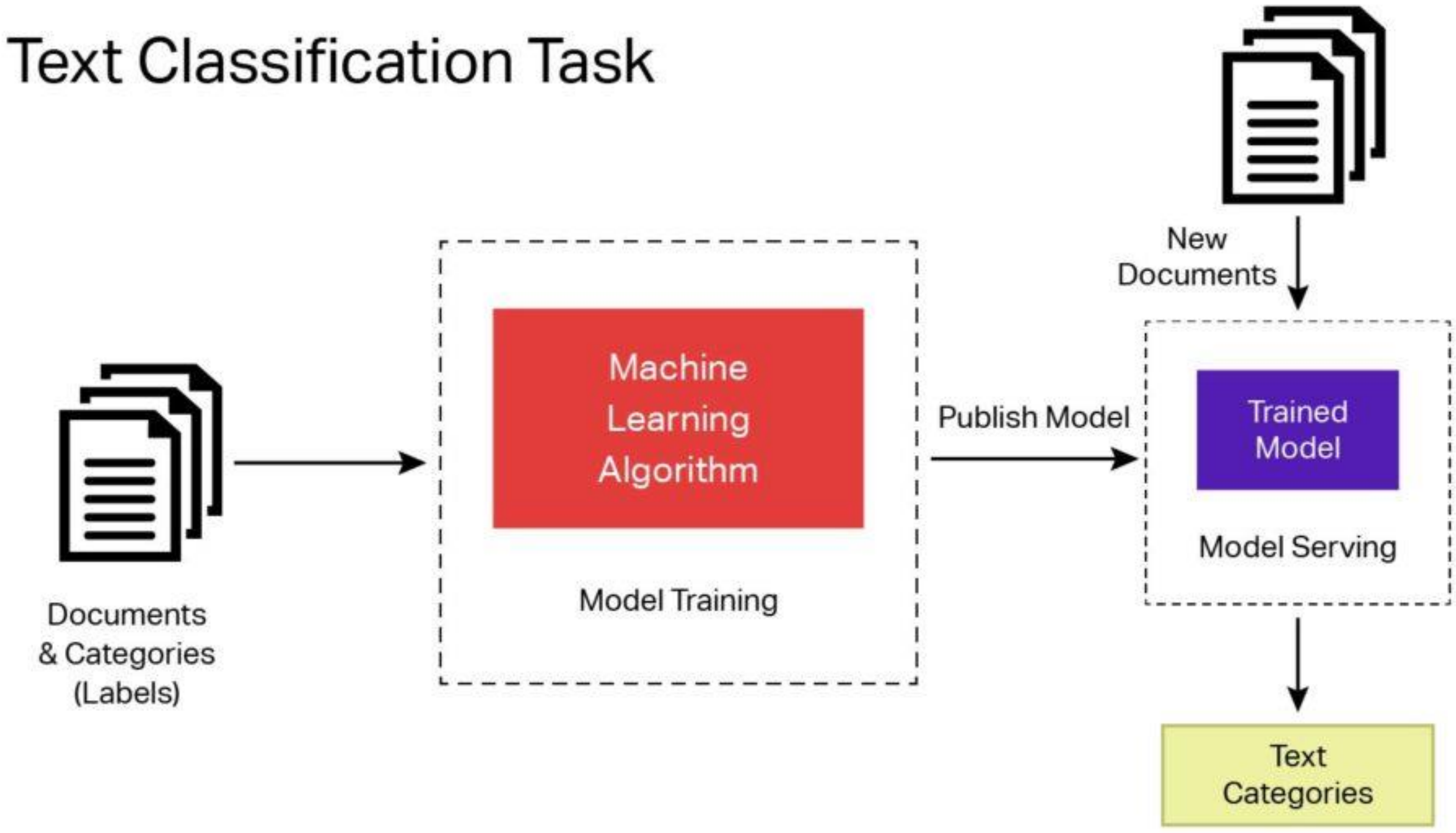
- Text classification involves **assigning text documents to predefined categories**.

- Types of Text Classification:
  - **Binary Classification:** Assigns one of two labels (0/1) (e.g., spam vs. not spam).

  - **Multiclass Classification:** Assigns one label from multiple categories (n classes) (e.g., classifying news into finance, weather, politics, etc.).

  - **Multilabel Classification:** Assigns multiple labels to a single document (e.g., tagging a movie as both "comedy" and "romance").

General Approach



https://imerit.net/blog/23-best-text-classification-datasets-for-machine-learning-all-pbm/

Formally

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Given a training sample (X, y), where X is a set of documents with corresponding labels y, from a set Y of possible labels, the task is to learn a function $f(x)$ that can predict the class $y' = f(x)$ for na unseen document (X, y).

Text Classification

- **Feature-based approaches** ("traditional" machine learning);
  - Rely on **handcrafted features** extracted from text, such as:
    - Bag-of-Words;
    - TF-IDF;
    - N-grams;
    - Linguistic features;
  - These approaches often use classifiers like Naive Bayes, Support Vector Machines (SVM), or Logistic Regression.

- **Neural approaches** (deep learning).
  - Leverage **neural networks to automatically learn features** from text, often involving:
    - Word embeddings (e.g., Word2Vec, GloVe);
    - Recurrent Neural Networks;
    - Transformers (e.g., BERT, GPT)
  - These methods allow for deeper, more context-aware text representations.

Text Classification Example

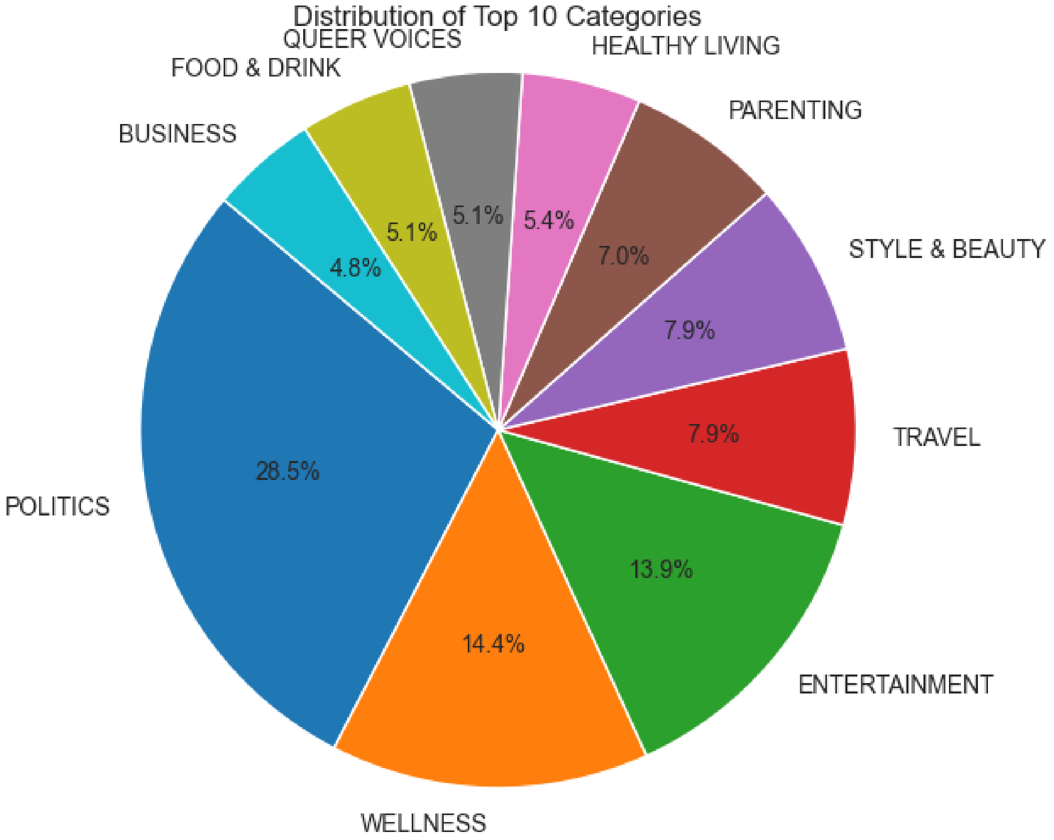
- Assume we have the following dataset:

	link	headline	category	short_description	authors	date
0	https://www.huffpost.com/entry/covid-boosters-...	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...	Carla K. Johnson, AP	2022-09-23
1	https://www.huffpost.com/entry/american-airlin...	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...	Mary Papenfuss	2022-09-23
2	https://www.huffpost.com/entry/funniest-tweets...	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...	Elyse Wanshel	2022-09-23
3	https://www.huffpost.com/entry/funniest-parent...	The Funniest Tweets From Parents This Week (Se...	PARENTING	"Accidentally put grown-up toothpaste on my to...	Caroline Bologna	2022-09-23
4	https://www.huffpost.com/entry/amy-cooper-lose...	Woman Who Called Cops On Black Bird-Watcher Lo...	U.S. NEWS	Amy Cooper accused investment firm Franklin Te...	Nina Golgowski	2022-09-22

Text Classification Example

- Assume we have the following dataset:

```
 Top 10 categories: Index(['POLITICS', 'WELLNESS', 'ENTERTAINMENT', 'TRAVEL', 'STYLE & BEAUTY',
       'PARENTING', 'HEALTHY LIVING', 'QUEER VOICES', 'FOOD & DRINK',
       'BUSINESS'],
      dtype='object', name='category')
category
POLITICS          35602
WELLNESS          17945
ENTERTAINMENT     17362
TRAVEL             9900
STYLE & BEAUTY     9814
PARENTING          8791
HEALTHY LIVING     6694
QUEER VOICES       6347
FOOD & DRINK       6340
BUSINESS           5992
Name: count, dtype: int64
```



Text Classification Example

- We need to **pre-process** our dataset to remove unwanted/ irrelevant parts (short description field).

```
Original Text: "Accidentally put grown-up toothpaste on my toddler's toothbrush and he screamed like I was cleaning his teeth with a Carolina Reaper dipped in Tabasco sauce."
```

  - Make it lowercase, removing text in square brackets,removing links, removing punctuation, and removing words containing numbers, etc;

```
Cleaned Text: accidentally put grownup toothpaste on my toddler's toothbrush and he screamed like i was cleaning his teeth with a carolina reaper dipped in tabasco sauce
```

  - Remove stop words;

```
Text without Stopwords: accidentally put grownup toothpaste toddler's toothbrush screamed like cleaning teeth carolina reaper dipped tabasco sauce
```

  - Apply stemming/lemmatization;

```
Stemmed Text: accident put grownup toothpast on my toddler toothbrush and he scream like i was clean his teeth with a carolina reaper dip in tabasco sauc
```

  - Etc.

```
Preprocessed Text: accident put grownup toothpast toddler toothbrush scream like clean teeth carolina reaper dip tabasco sauc
```

Text Classification Example

- Convert text data into numerical vectors:
    - **TF-IDF**;
    - Bag-of-Words;
    - N-grams;
    - Etc.

```
>>> from sklearn.feature_extraction.text import TfidfVectorizer
>>> corpus = [
...     'This is the first document.',
...     'This document is the second document.',
...     'And this is the third one.',
...     'Is this the first document?',
... ]
>>> vectorizer = TfidfVectorizer()
>>> X = vectorizer.fit_transform(corpus)
>>> vectorizer.get_feature_names_out()
array(['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third',
       'this'], ...)
>>> print(X.shape)
(4, 9)
```

Text Classification Example

- Train a model

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(random_state=42)
model.fit(tfid_X_train, tfid_y_train)
```

- Evaluate the model

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

y_pred = model.predict(tfid_X_test)
print(f'Predicted values: {y_pred[:5]}')
acc = accuracy_score(tfid_y_test, y_pred)
print(f'Accuracy: {acc}')
precision = precision_score(tfid_y_test, y_pred, average='weighted')
print(f'Precision: {precision}')
# Calculate recall
recall = recall_score(tfid_y_test, y_pred, average='weighted')
print(f'Recall: {recall}')
```

Text Classification Evaluation

- Assessing Classifier Performance:
  - Compare predicted labels (y') with true labels (y) for each document in a dataset.
- Confusion Matrix:
  - Summarizes correct and incorrect predictions in a table.
- Metrics:
  - Use the confusion matrix to calculate performance metrics like accuracy, precision, recall, and F1-score.

Evaluating Binary Predictions

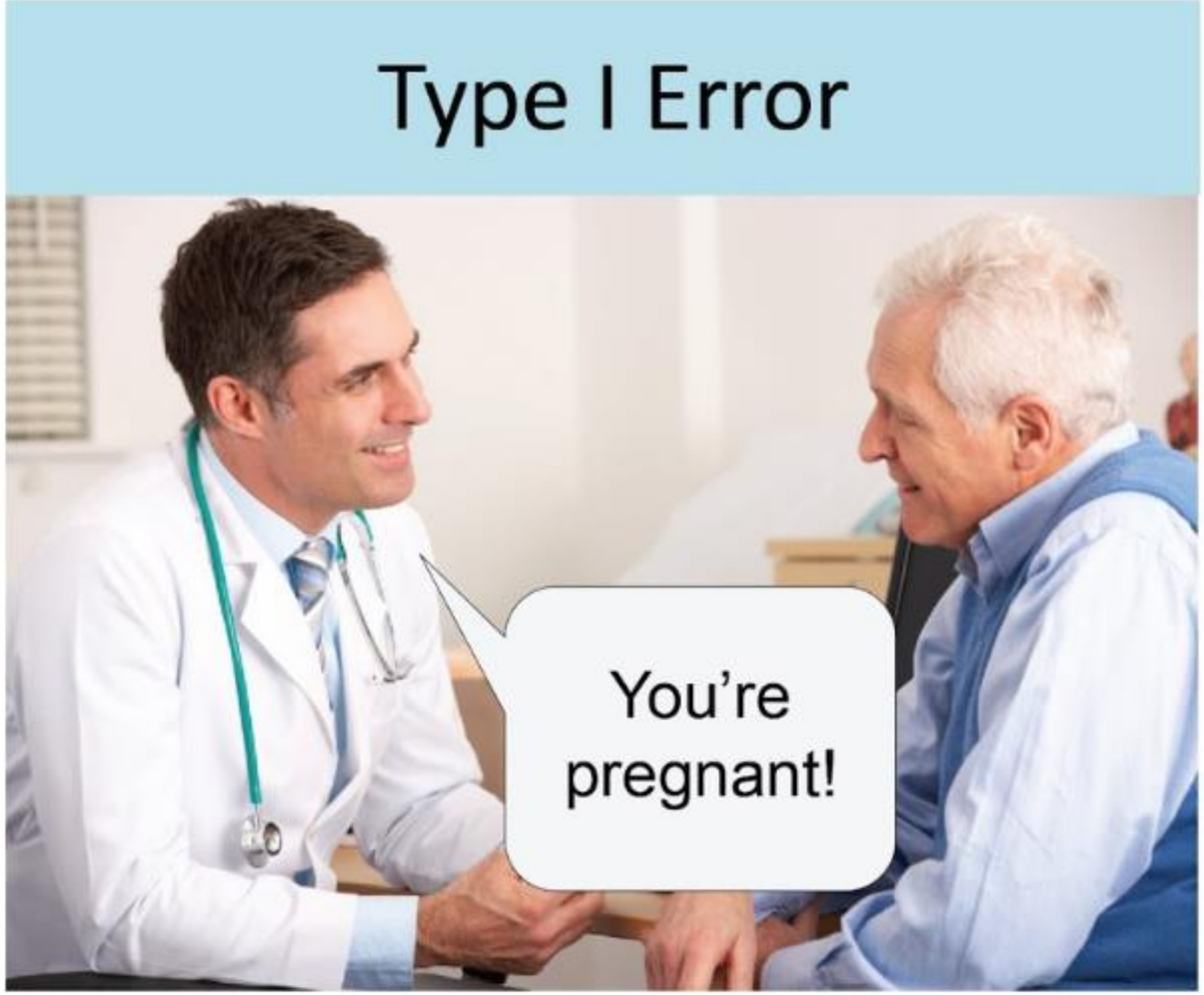
UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

		Predicted class	
		negative	positive
Actual class	negative	true negatives (TN)	false positives (FP)
	positive	false negatives (FN)	true positives (TP)

- False positives = Type I error ("raising a false alarm")
- False negatives = Type II error ("failing to raise an alarm")

- **Which of Type I or Type II error is worse?**

Type I vs. Type II Errors



https://www.analyticsindiamag.com/understanding-type-i-and-type-ii-errors/

Confusion Matrix Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-		
	+		

Confusion Matrix Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-	**2**	
	+		

Confusion Matrix Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

predicted

actual		-	+
	-	2	**1**
	+		

Confusion Matrix Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

actual \ predicted	-	+
-	2	1
+	4	3

Evaluation Measures

- Summarizing performance in a single number

- **Accuracy**
    - Fraction of correctly classified items out of all items

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

predicted

		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **Error rate**
  - Fraction of incorrectly classified items out of all items

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **Precision**
  - Fraction of items correctly identified as positive out of the total items identified as positive.

$$P = \frac{TP}{TP + FP}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **Recall** (also called Sensitivity or True Positive Rate)
  - Fraction of items correctly identified as positive out of the total actual positives

$$R = \frac{TP}{TP + FN}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **F1-Score**
    - The harmonic mean of precision and recall

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

		predicted -	predicted +
actual	-	TN	FP
actual	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **False Positive rate (Type I Error)**
  - Fraction of items wrongly identified as positive out of the total actual negatives

$$FPR = \frac{FP}{FP + TN}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation Measures

- Summarizing performance in a single number

- **False Negative Rate (Type II Error)**
    - Fraction of items wrongly identified as negative out of the total actual positives

$$FNR = \frac{FN}{FN + TP}$$

		predicted -	predicted +
actual	-	TN	FP
actual	+	FN	TP

Metrics Example

		predicted -	predicted +
actual	-	TN=2	FP=1
actual	+	FN=4	TP=3

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5}{10} = 0.5$$

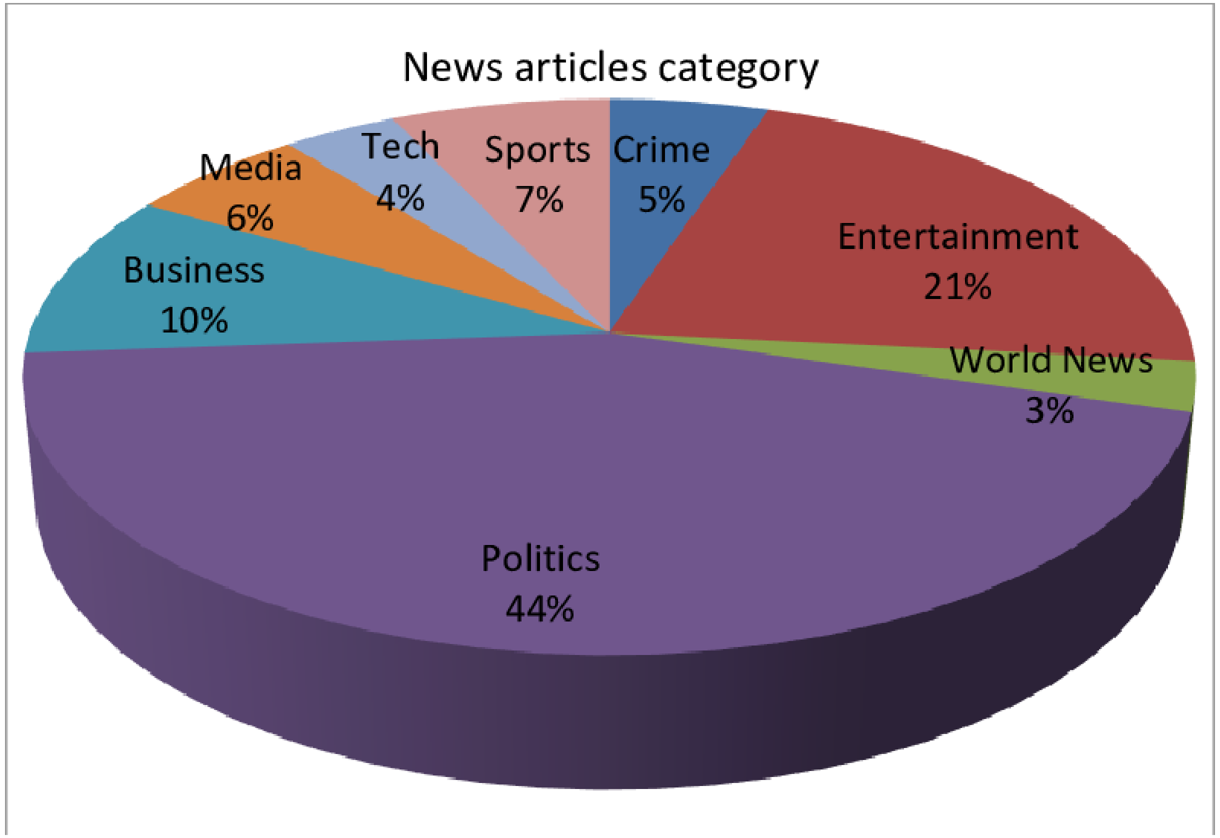
$$P = \frac{TP}{TP + FP} = \frac{3}{4} = 0.75$$

$$R = \frac{TP}{TP + FN} = \frac{3}{7} = 0.429$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot {}^3/_4 \cdot {}^3/_7}{{}^3/_4 + {}^3/_7} = 0.545$$

Multiclass Classification

- Imagine that we need to automatically sort news stories according to their topical categories.



Multiclass Classification

- Many algorithms are originally built for binary classification.

- Sometimes, those can be adapted for multiclass:
  - **One-vs-Rest**: Train one classifier per class vs. all others.
  - **One-vs-One**: Train classifiers for every pair of classes.

  - **Voting Scheme**: Combine predictions through voting, with tie-breaking mechanisms if needed.

One-vs-Rest

- **Setup:**
  - For k target classes (y_1, ..., y_k), train one classifier per class.

- **Training:**
  - Class y_i: positive examples.
  - All other classes (yj, $j \neq i$): negative examples.

- **Combining Predictions:**
  - Each classifier votes for its class if it predicts positive.
  - The class with the most votes wins.

One-vs-Rest

- 4 classes (y_1, y_2, y_3, y_4)
- Classifying a given test intance (dots indicate the votes cast):

y_1	+	•
y_2	-	
y_3	-	
y_4	-	
Pred.	+	

y_1	-	•
y_2	+	
y_3	-	•
y_4	-	•
Pred.	-	

y_1	-	•
y_2	-	•
y_3	+	
y_4	-	•
Pred.	-	

y_1	-	•
y_2	-	•
y_3	-	•
y_4	+	
Pred.	-	

- Sum votes received: (y_1, ••••) (y_2, ••) (y_3, ••) (y_4, ••)

One-vs-One

- **Setup:**
  - For k target classes (y_1, ..., y_k), build a binary classifier for each pair (y_1, ..., y_k).
- **Number of classifiers:**
  - A total of $k \cdot (k - 1) / 2$ classifiers.
- **Combining Predictions:**
  - Each pairwise comparison gives a vote to the predicted class.
  - The class with the most votes wins.

One-vs-One

- 4 classes (y_1, y_2, y_3, y_4)

- Classifying a given test instance (dots indicate the votes cast):

y_1	+	•
y_2	-	
Pred.	+	

y_1	+	•
y_3	-	
Pred.	+	

y_1	+	
y_4	-	•
Pred.	-	

y_2	+	•
y_3	-	
Pred.	+	

y_2	+	
y_4	-	•
Pred.	-	

y_3	+	•
y_4	-	
Pred.	+	

- Sum votes received: (y_1, ••) (y_2, •) (y_3, •) (y_4, ••)

Evaluating Multiclass Classification

- Accuracy can still be computed as:

$$ACC = \frac{\#\text{correctly classified instances}}{\#\text{total number of instances}}$$

- For other metrics:
  - View it as a set of k binary classification problems (k is the number of classes)
  - Create confusion matrix for each class by evaluating "one agains the rest"
  - Average over all classes

Confusion Matrix

		Predicted				
		1	2	3	...	k
Actual	1	**24**	0	2		0
	2	0	**10**	1		1
	3	1	0	**9**		0
	...					
	k	2	0	1		**30**

Binary Confusion Matrices (One-Against-Rest)

		Predicted				
		1	2	3	...	k
Actual	1	**24**	0	2		0
	2	0	**10**	1		1
	3	1	0	**9**		0
	...					
	k	2	0	1		**30**

$\Rightarrow$

		Predicted	
		1	$\neg$1
Act.	1	TP=24	FN=3
	$\neg$1	FP=2	TN=52

		Predicted	
		2	$\neg$2
Act.	2	TP=10	FN=2
	$\neg$2	FP=0	TN=69

...

For the sake of this illustration, we assume that the cells which are not shown are all zeros.

Averaging over classes

- Averaging can be performed on the instance level or on the class level.

- **Micro-averaging** aggregates the results of individual instances across all classes.
  - All instances are treated equal

- **Macro-averaging** computes the measure independently for each class and then take the average.
  - All classes are treated equal

Micro-Averaging

- Precision

$$P_\mu = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k}(TP_i + FP_i)}$$

- Recall

$$R_\mu = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k}(TP_i + FN_i)}$$

- F1-Score

$$F1_\mu = \frac{2 \cdot P_\mu \cdot R_\mu}{P_\mu + R_\mu}$$

		predicted	
		i	$\neg i$
actual	i	TP_i	FN_i
	$\neg i$	FP_i	TN_i

Macro-Averaging

- Precision

$$P_M = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FP_i}}{k}$$

- Recall

$$R_M = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FN_i}}{k}$$

- F1-Score

$$F1_M = \frac{\sum_{i=1}^{k} \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}}{k}$$

		predicted	
		i	$\neg i$
actual	i	TP_i	FN_i
	$\neg i$	FP_i	TN_i

where P_i and R_i are Precision and Recall, respectively, for class i

Model Evaluation – Holdout Method

- **Idea:** hold out part of the training data for testing
  - This data is **not used during training**, allowing for an independent evaluation of model performance.
  - Helps **prevent overfitting** by providing an estimate of **how the model generalizes to unseen data**.

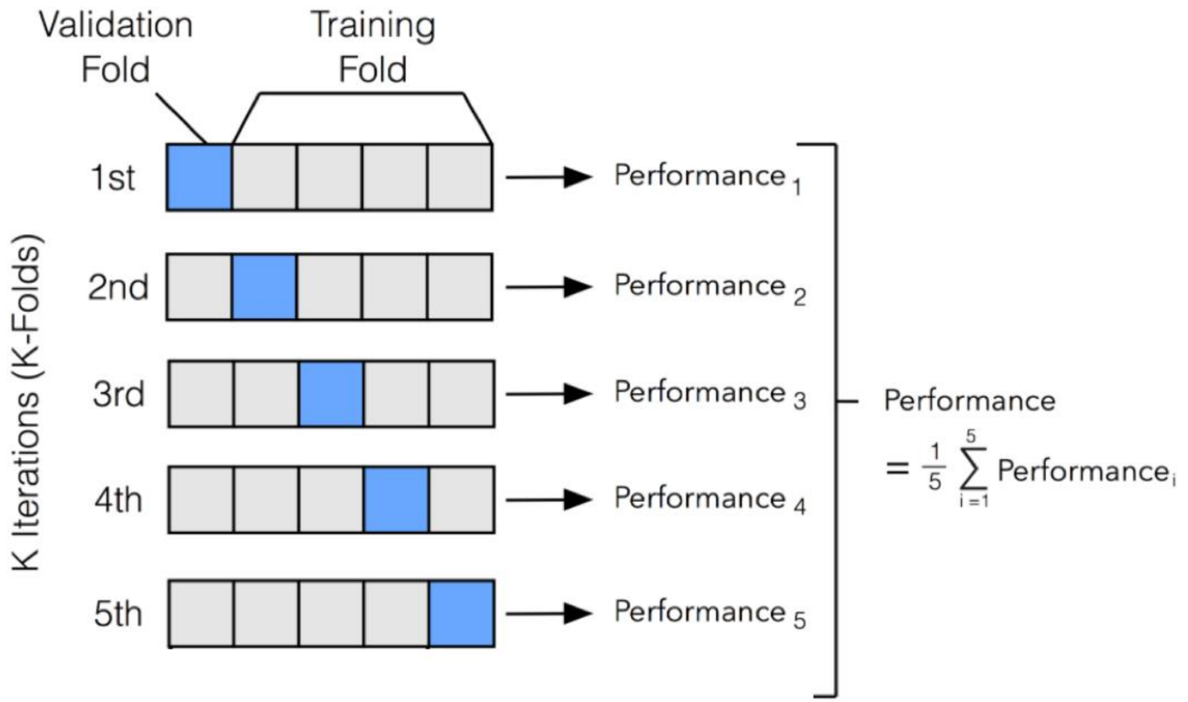
- **Single train/validation split**
  - Split the training data into X% training split and 100 – X% validation split (e.g., 80%/20%).

https://community.alteryx.com/t5/Data-Science/Holdouts-and-Cross-Validation-Why-the-Data-Used-to-Evaluate-your/ba-p/448982

Model Evaluation - *k*-fold Cross-Validation

- Divide the training data randomly into k **folds**;
- Use k**-1 folds for training** and **test on the** k**th fold**;
- **Repeat** k **times** (each fold is used for testing exactly once).

- Specific case when k **is equal to the the number of data points** is called: "**leave-one-out**" evaluation.



http://ethen8181.github.io/machine-learning/model_selection/model_selection.html