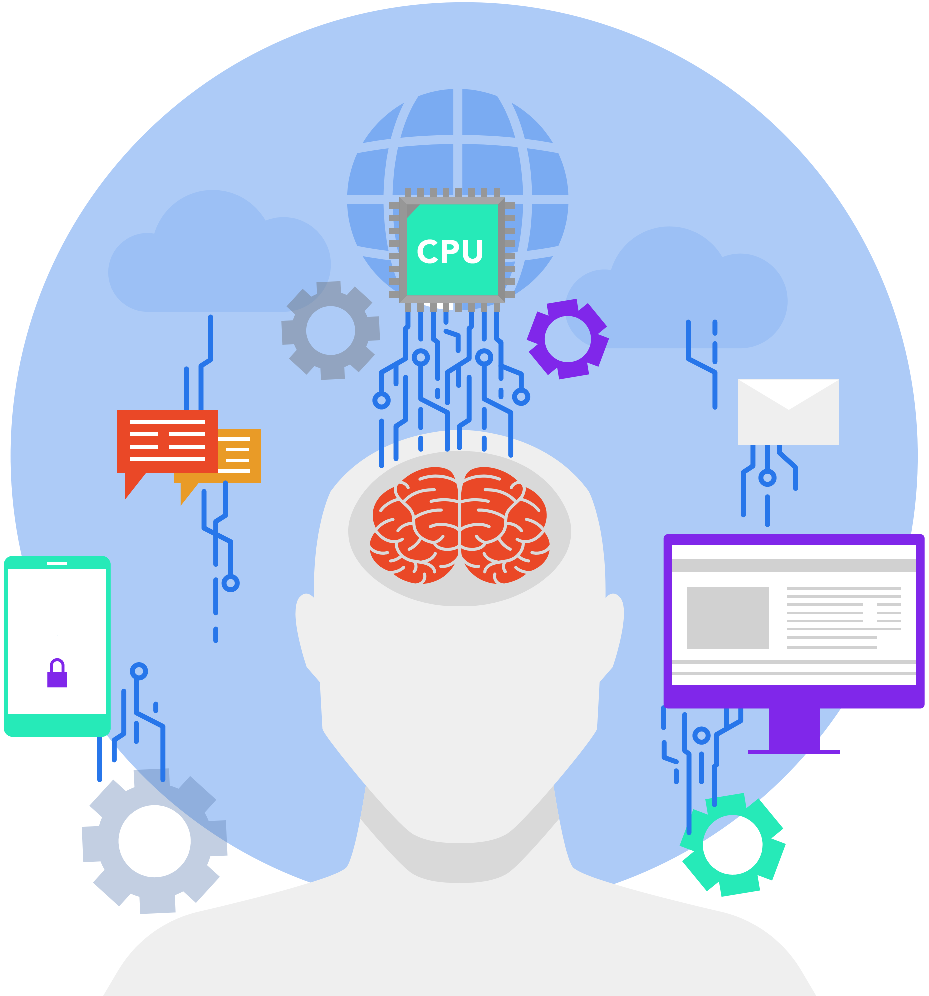


News Topic Classification

Build NLP classifiers
for a specific text

Presentation Route

01 Introduction

02 Data Overview

03 Exploratory Data Analysis

04 Pre-processing

05 Classifier Selection

06 Model Evaluation

07 Error Analysis

08 Conclusion

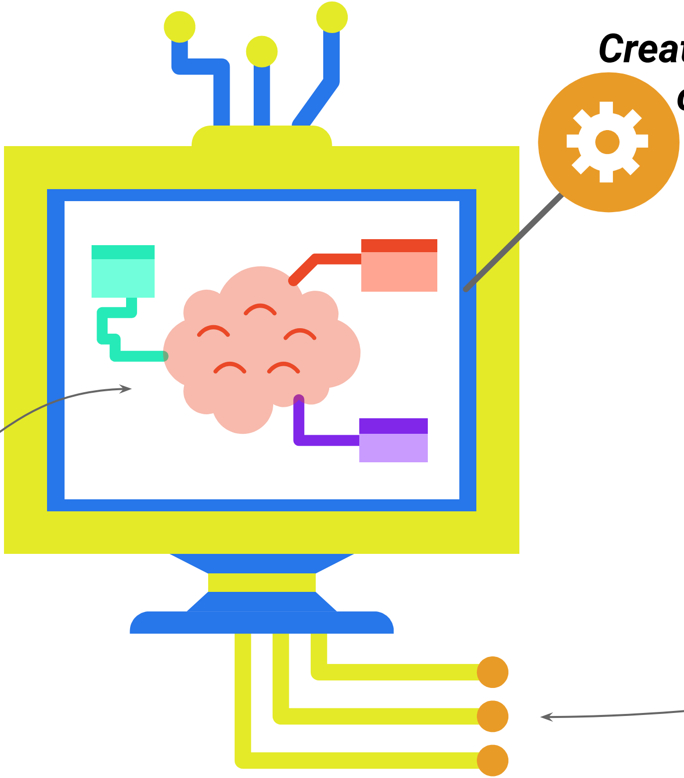
09 Q&A

Aim of the project

Inputs

Given news articles text

- **New 1**
- **New 2**
- **New 3**



Create NLP classifiers

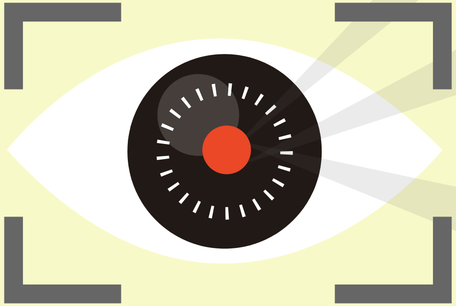
Outputs

to label as 0, 1, 2 or 3

- **Label 1**
- **Label 2**
- **Label 3**

Data Overview

Understanding the dataset



Source

Downloaded from **Kaggle**
Gathered from > 2000 news sources by ***ComeToMyHead***

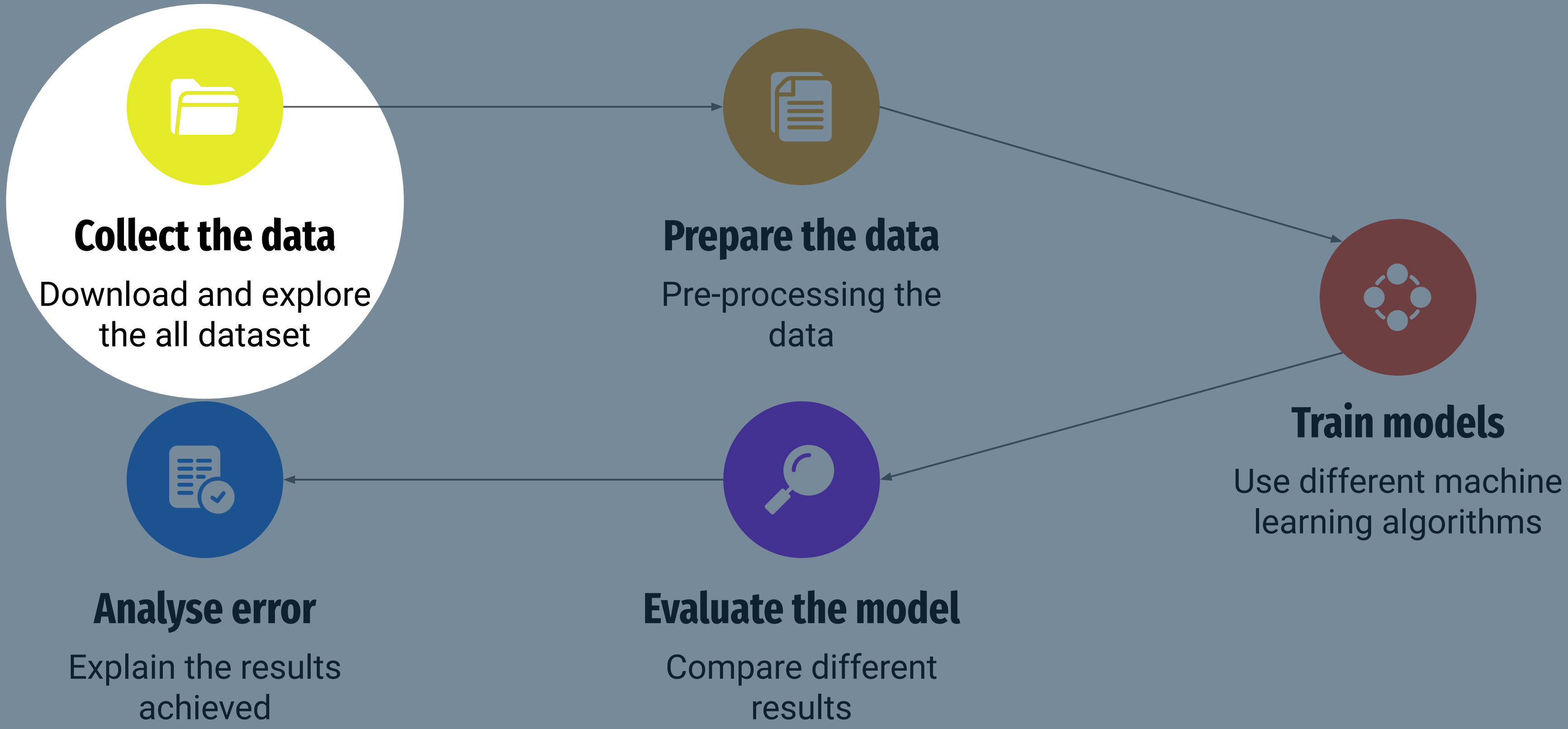
Properties

Size: **29MB**; Language: **ENG**
2 columns: ***Text*** and ***Label***
4 types of news topics: *'World'*, *'Sports'*, *'Business'* and *'Sci/Tech'*.

Data Quality

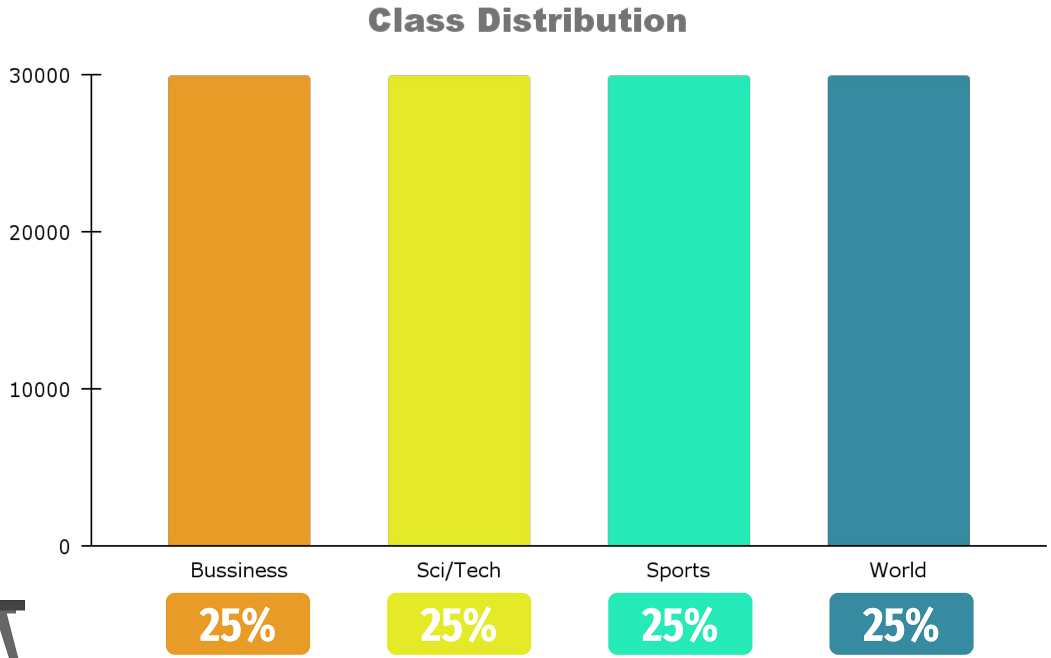
No missing values
Last update **2 months ago**

Project Pipeline

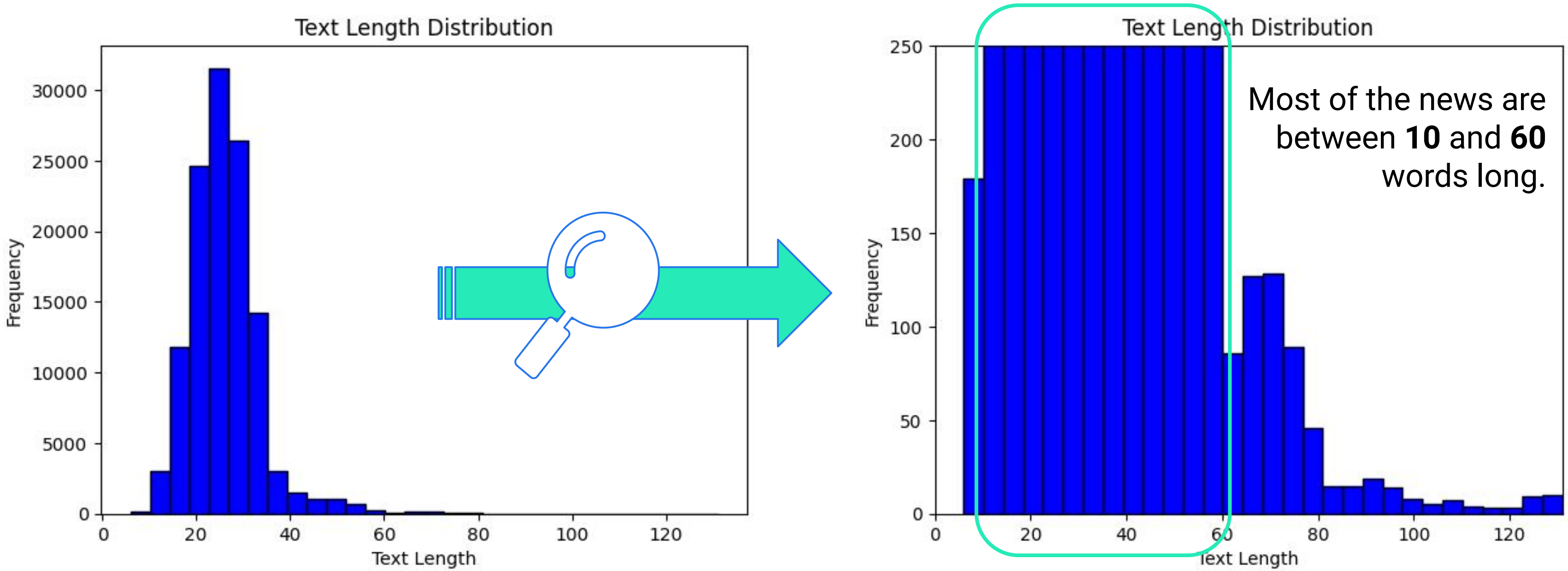


Data Exploration

The dataset is **perfectly balanced**, as each label has the same amount of samples, exactly **30000 news** per type.

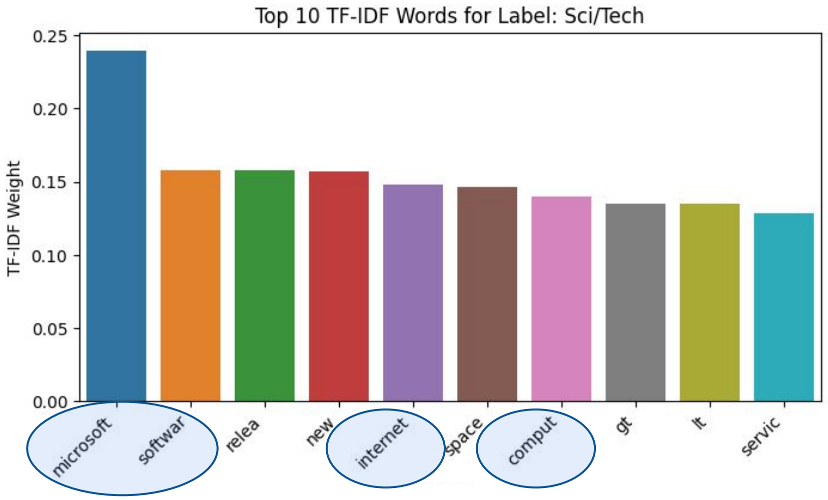


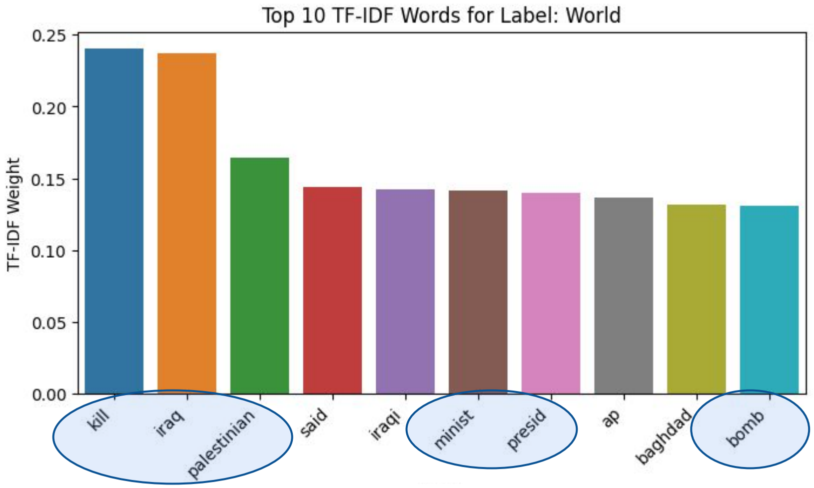
Data Exploration



Most of the news are between **10** and **60** words long.

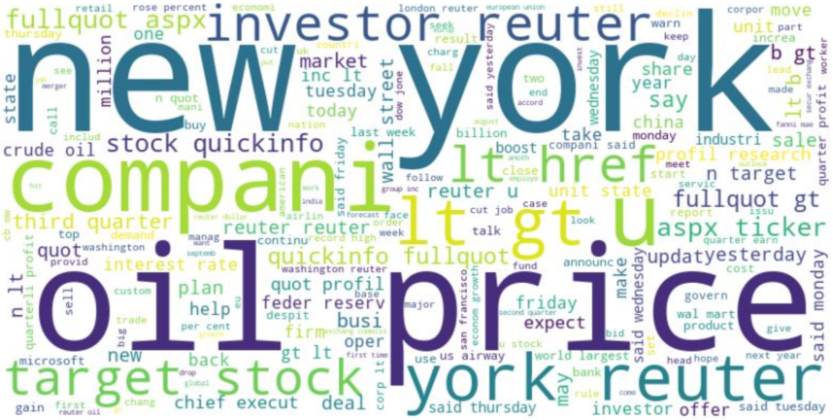
Data Exploration





Data Exploration

Word Cloud for Class: Business



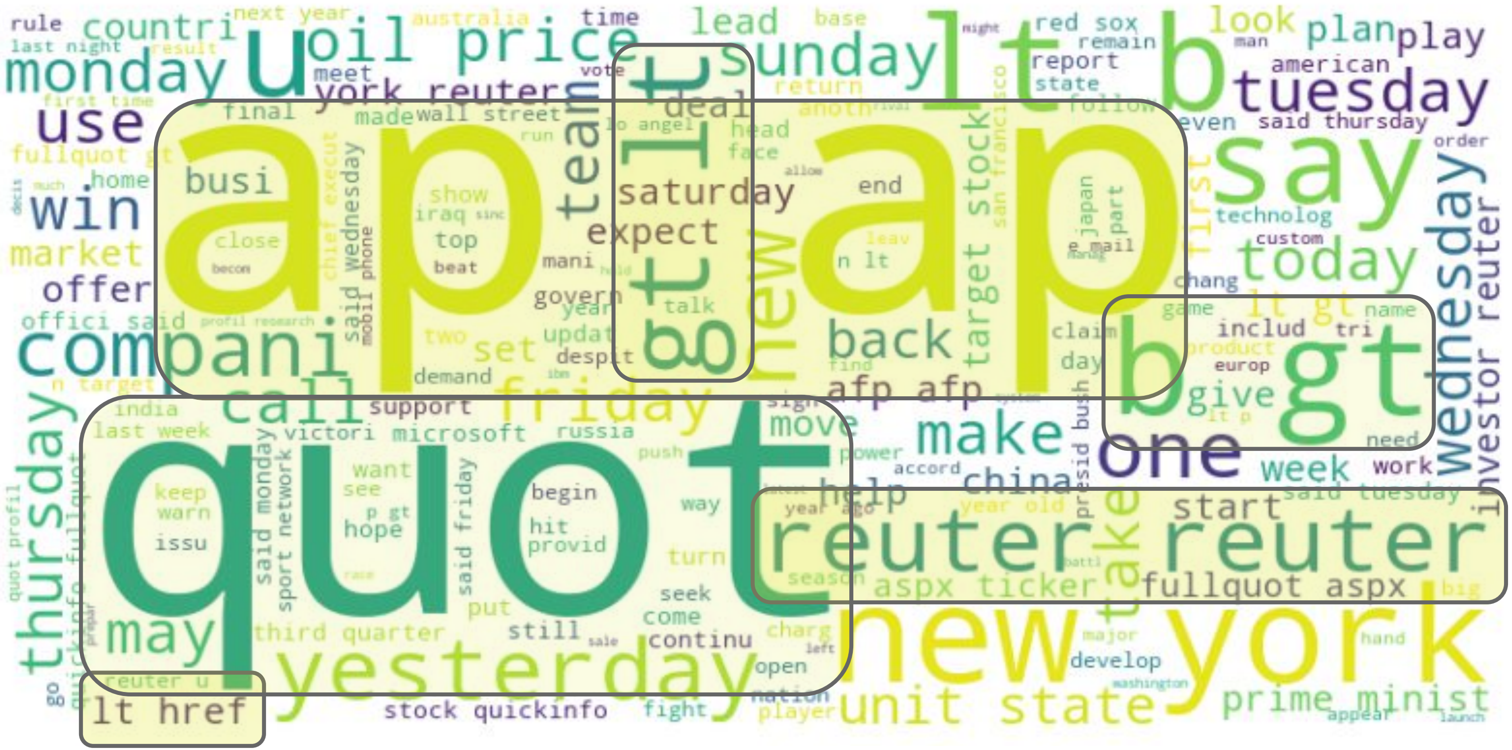
Word Cloud for Class: Sports

- *New York*
- *Oil Price*
- *Target Stock*
- *Investor*

- *Win*
- *Team*
- *Game*
- *Play*

Data Exploration

Word Cloud of all the dataset



Project Pipeline

Collect the data

Download and explore the all dataset

Prepare the data

Pre-processing the data

Train models

Use different machine learning algorithms

Evaluate the model

Compare different results

Analyse error

Explain the results achieved

Data Preparation

Loading the pre-trained English language model

```
import spacy
nlp = spacy.load("en_core_web_md")
```

01

Sampled Text

Sampled one third of the data, keeping it balanced (10.000 for each label)

02

Clean the text

Use regular expressions to remove links, html tags, reuters and quotes

03

Apply spaCy pipeline

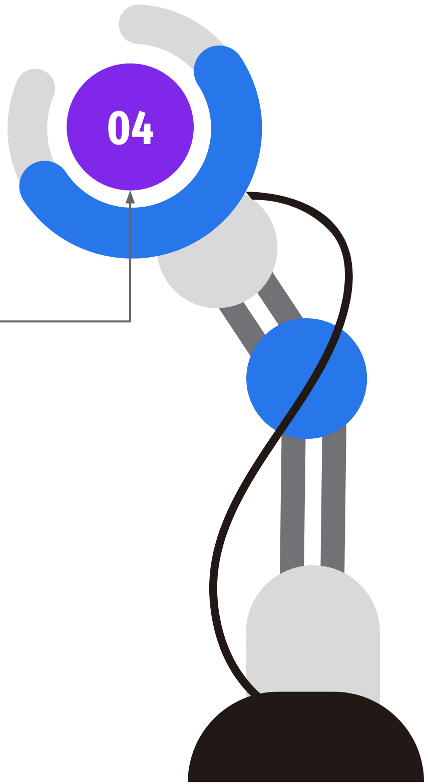
Applies the *spaCy* NLP pipeline to each entry in the 'text' column

```
df['text'].apply(nlp)
```

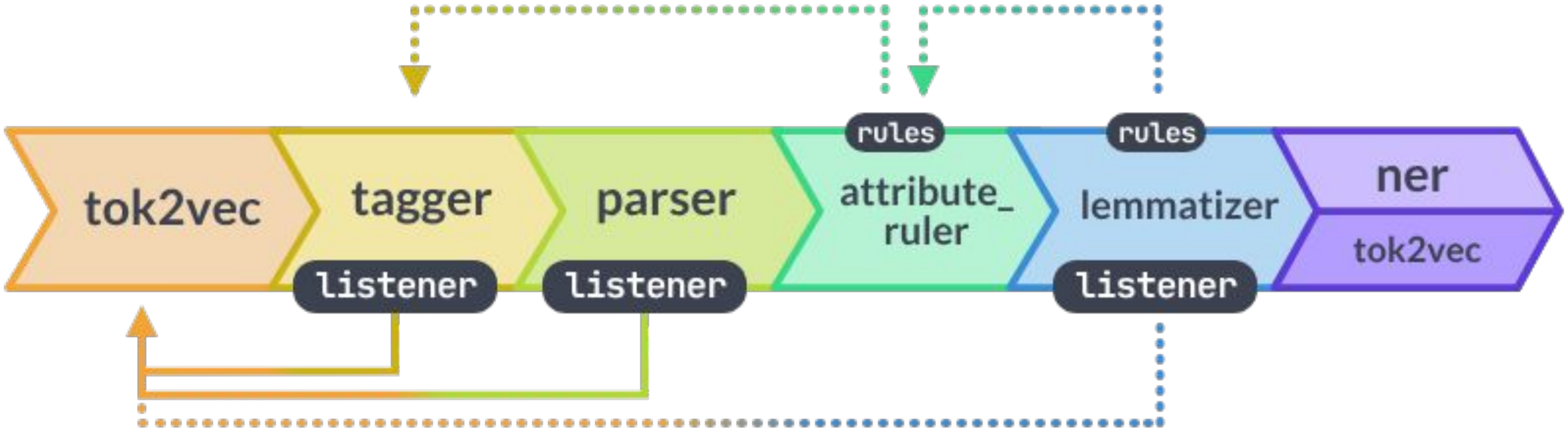
04

Feature Extration

Used the class *Token* generated by *spaCy* to create new features to use in the models



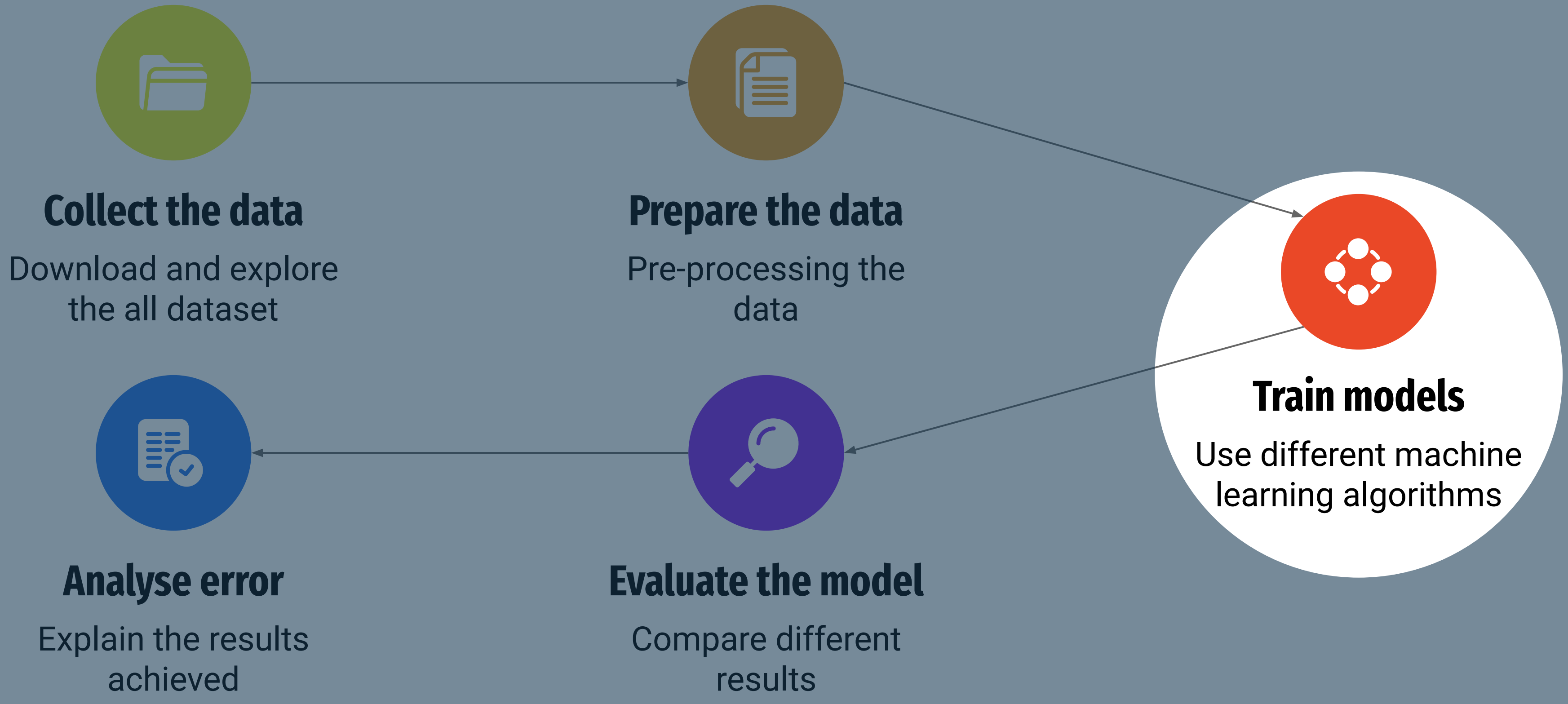
spaCy Pipeline Used



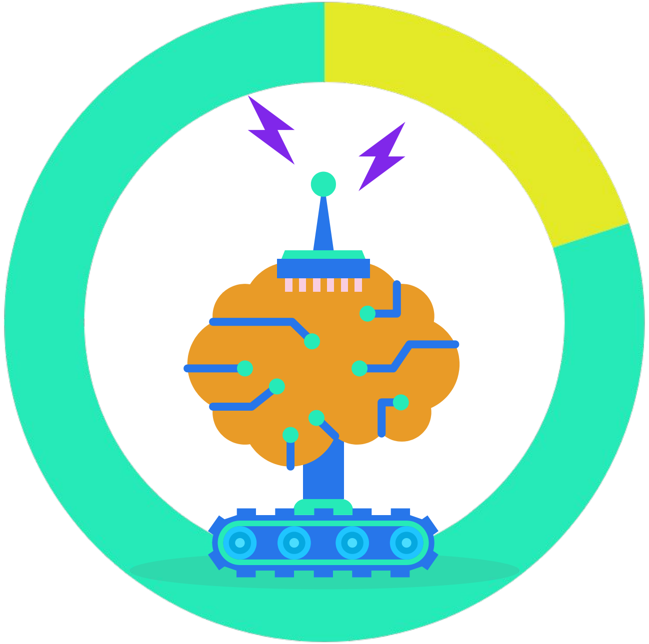
Data Preparation

	text	label	tokens	tokens_count	tokens_filtered	text_filtered	text_embeddings	text_ner	entity_dict
568	Nearly 10 Million Afghans to Embrace Democracy...	0	(Nearly, 10, Million, Afghans, to, Embrace, De...	43	[Nearly, 10, Million, Afghans, Embrace, Democr...	nearly 10 million afghans embrace democracy re...	[[-1.2955, -2.7019, -1.6919, 1.7825, 5.9797, 1...	[(Nearly, ADV, B, CARDINAL), (10, NUM, I, CARD...	{'nearly 10 million': ('NUM', 'CARDINAL'), 'af...
320	Lenovo revenue grows, but problems persist Chi...	3	(Lenovo, revenue, grows, ,, but, problems, per...	24	[Lenovo, revenue, grows, problems, persist, Ch...	lenovo revenue grow problem persist china larg...	[[0.92553, 2.4457, -0.12281, 3.1267, 0.7986, 2...	[(Lenovo, PROPN, B, ORG), (revenue, NOUN, O,)...	{'china': ('PROPN', 'GPE')}
93	Bangkok's Canals Losing to Urban Sprawl (AP) A...	3	(Bangkok, 's, Canals, Losing, to, Urban, Spraw...	43	[Bangkok, Canals, Losing, Urban, Sprawl, AP, A...	bangkok canals lose urban sprawl ap ap bank ca...	[[-0.26372, -0.95107, -0.19456, 1.6077, 2.4591...	[(Bangkok, PROPN, B, GPE), (Canals, PROPN, O, ...	{'bangkok canal': ('VERB', 'ORG'),

Project Pipeline



Train-test Split



80%

Training set

The model learns patterns and relationships in the training data to make predictions.

20%

Testing set

The predicted outputs are then compared to the actual target labels in the testing set

Feature Sets

01 **features**

All features (entity_dict, word_embeddings, text_filtered, tokens_filtered)

02 **features_embedds**

Only embeddings (word_embeddings)

03 **features_text_and_tokens**

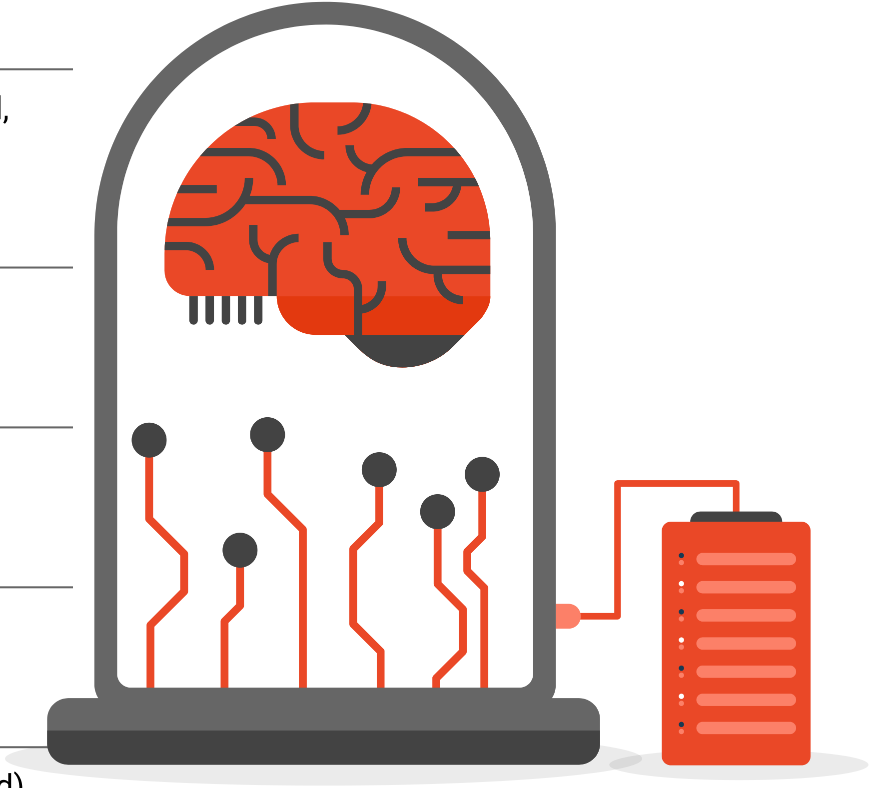
Only text and tokens (text_filtered, tokens_filtered)

04 **features_entities**

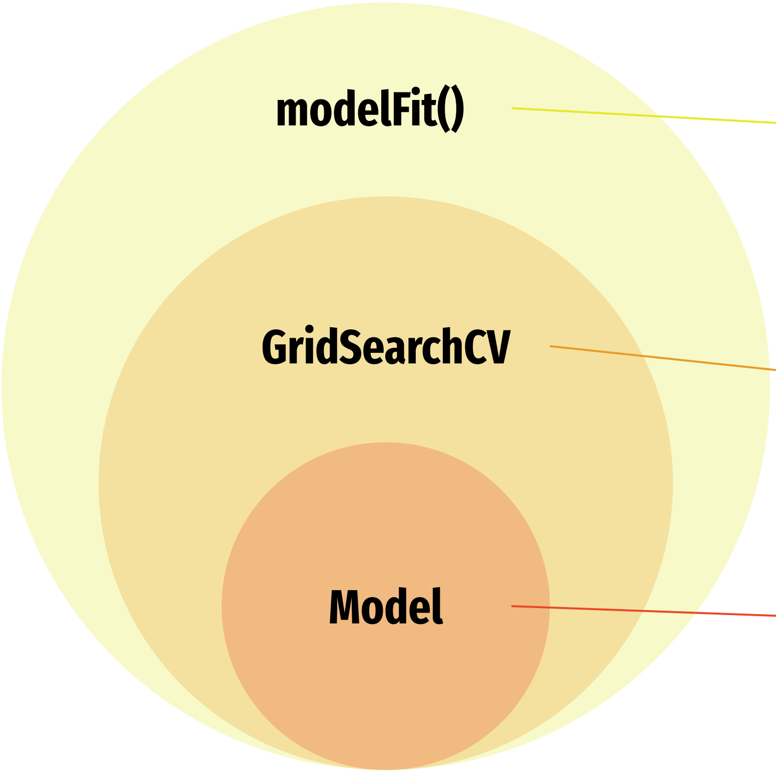
Only entities (entity_dict)

05 **features_no_ner**

No NER (word_embeddings, text_filtered, tokens_filtered)



Train a model

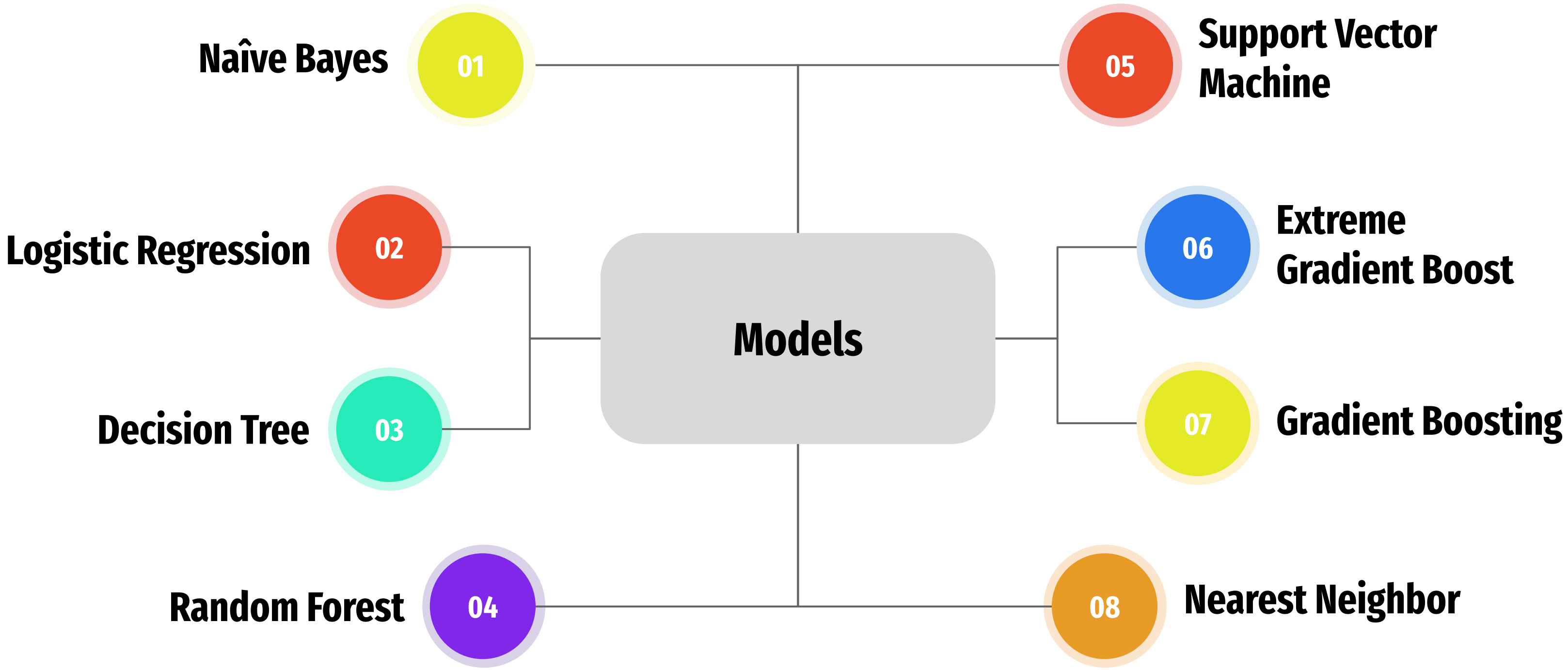


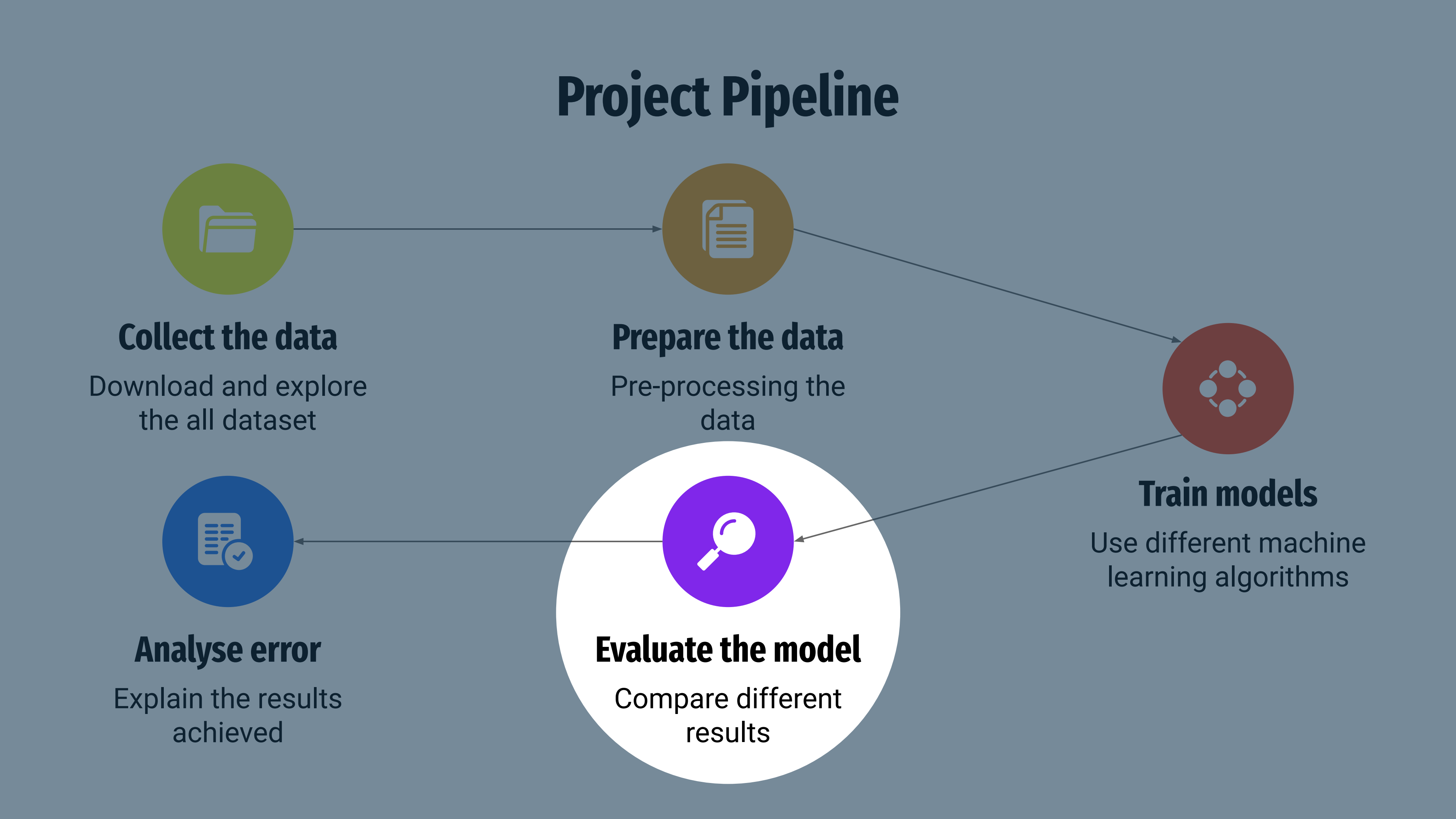
Function that facilitates the training and evaluation of an automatic learning model using GridSearchCV for hyperparameter tuning

Fits the GridSearchCV object on the training data and uses the best model to make predictions on the test data

The machine learning model to be trained and evaluated together with the grid of hyperparameters to be searched

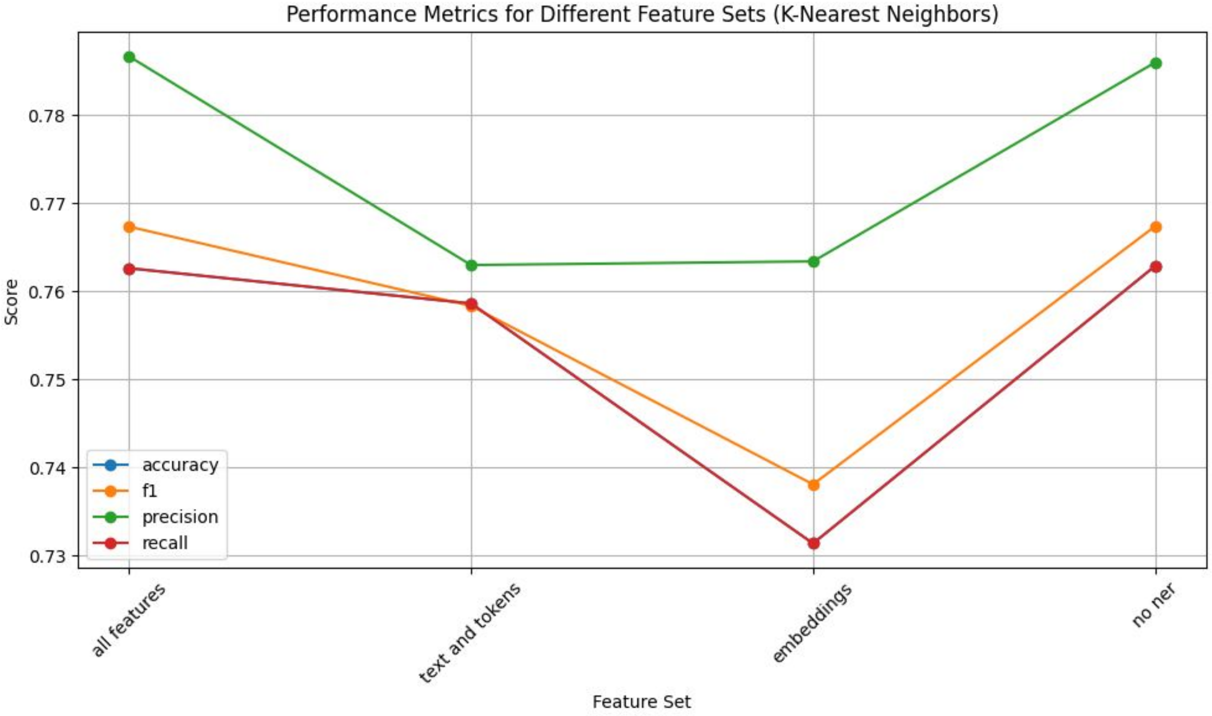
Train a model




Project Pipeline
Collect the data
Download and explore the all dataset
Prepare the data
Pre-processing the data
Train models
Use different machine learning algorithms
Evaluate the model
Compare different results
Analyse error
Explain the results achieved

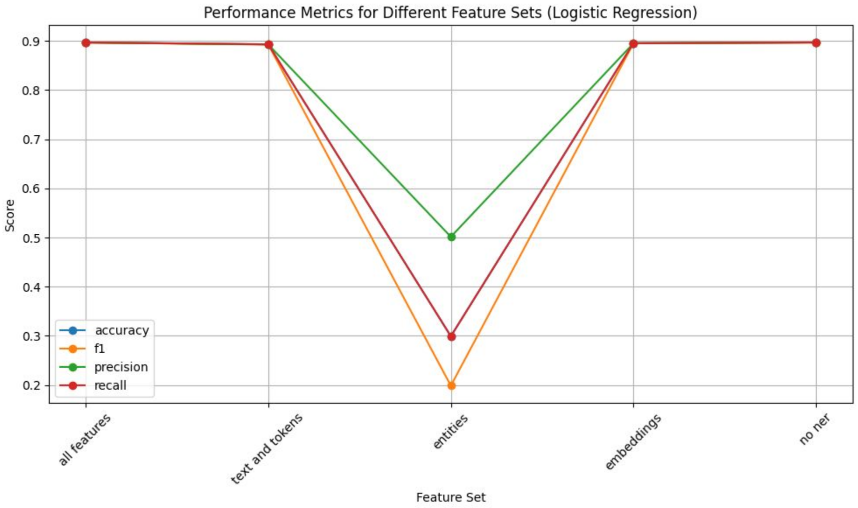
Evaluate the models	Accuracy	F1-Score	Precision	Recall
Naîve Bayes	0.895	0.894	0.894	0.895
Logistic Regression	0.897	0.896	0.896	0.897
Decision Tree	0.772	0.772	0.772	0.772
Random Forest	0.86	0.86	0.86	0.86
Support Vector Machine	0.896	0.895	0.895	0.896
Extreme Gradient Boost	0.88	0.88	0.88	0.88
Gradient Boosting	0.89	0.89	0.89	0.89
Nearest Neighbor	0.763	0.767	0.787	0.763

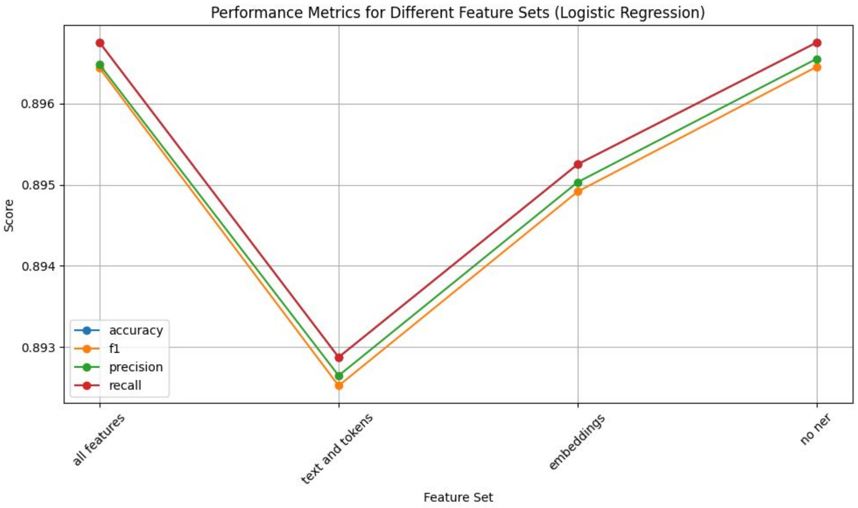
Metric Analysis



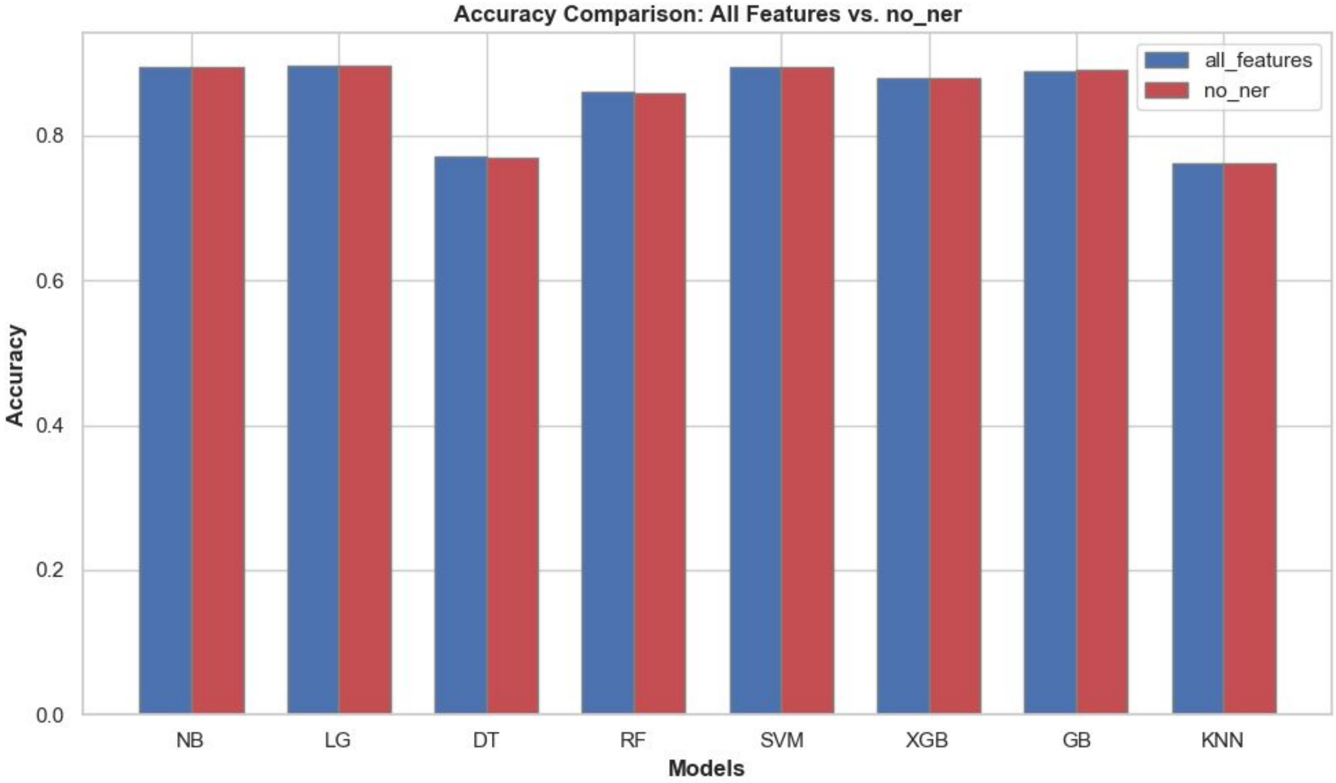
Metric Analysis

(Different feature sets for Logistic Regression)

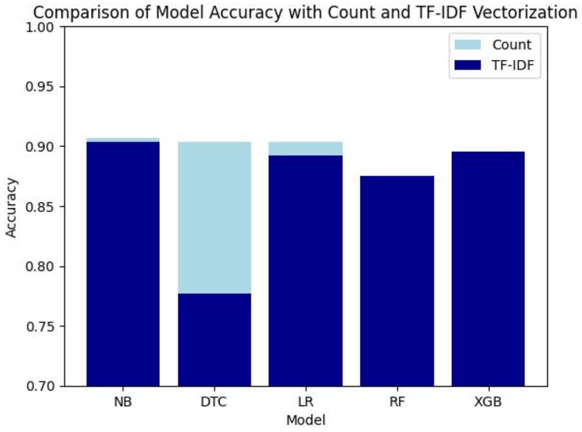


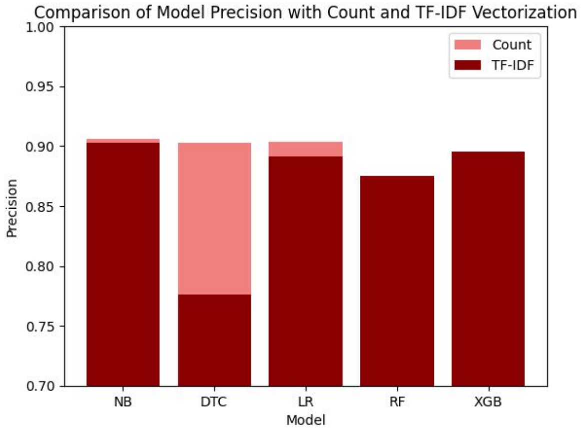


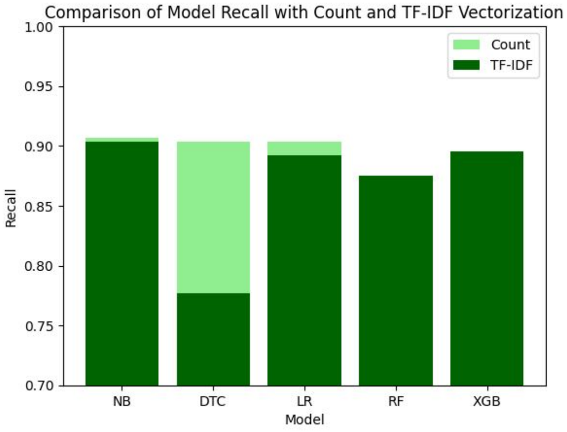
Metric Analysis

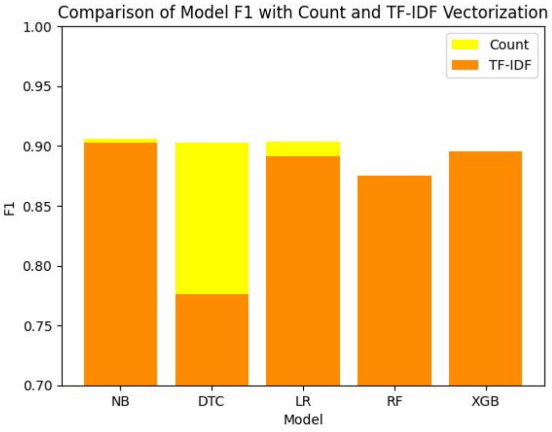


Count vs TF-IDF Vectorization

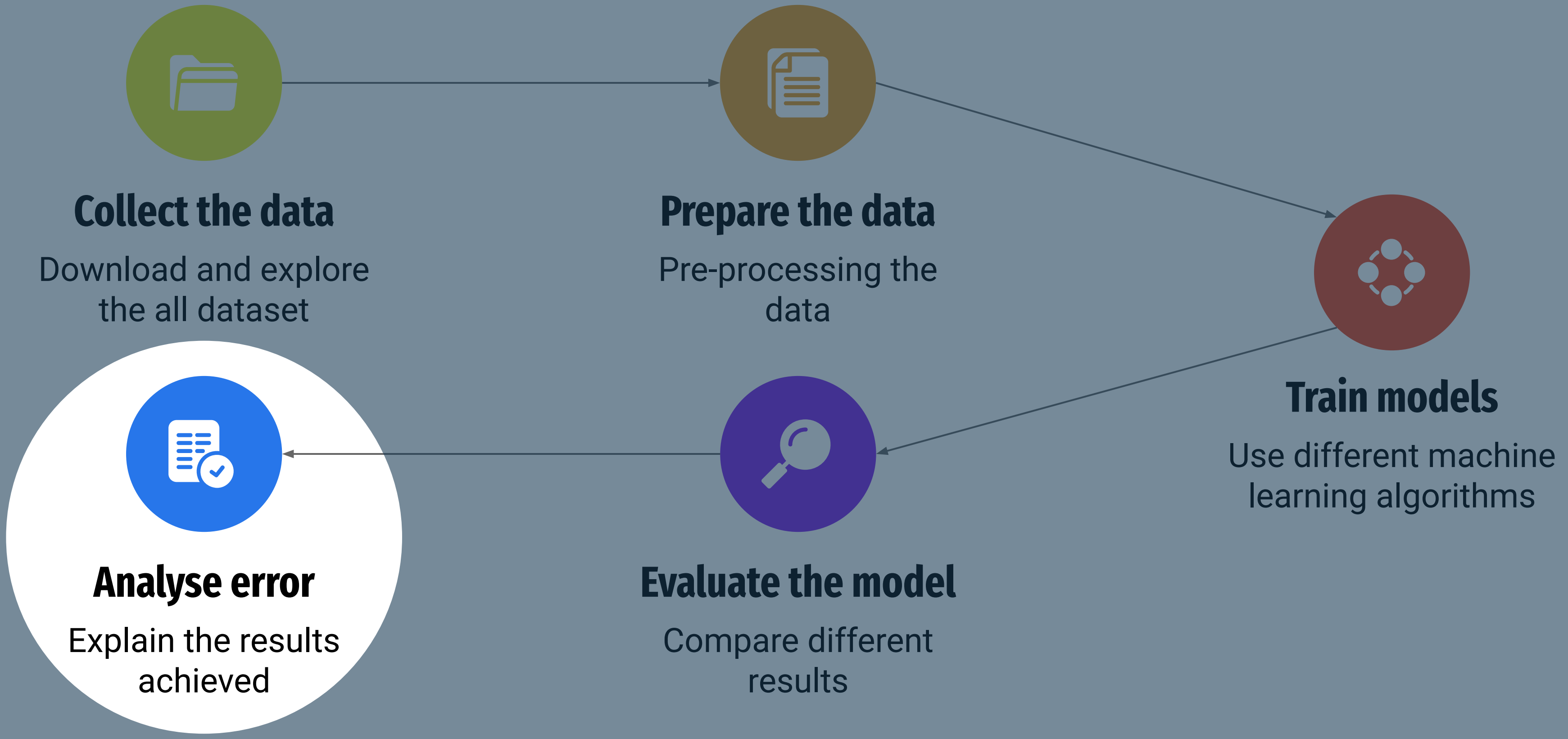








Project Pipeline

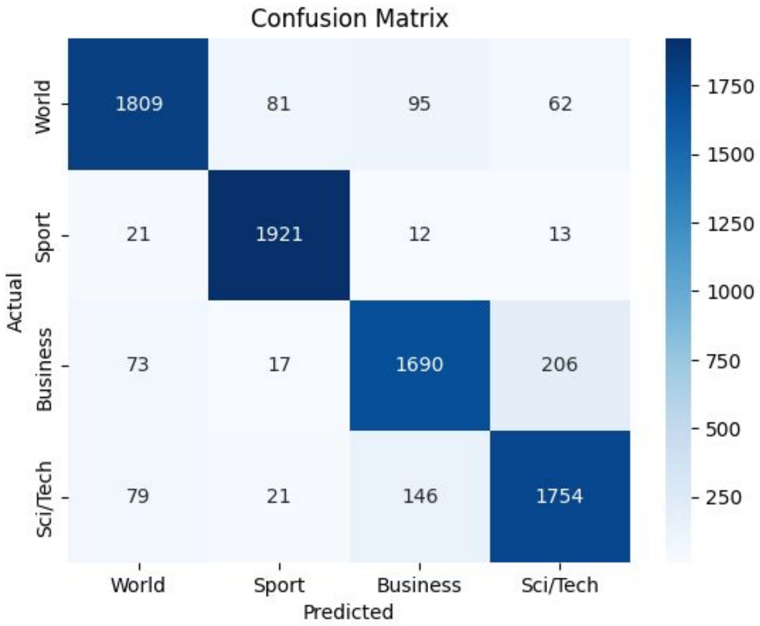


Logistic Regression *All Features*

Ambiguous Sci/Tech news articles examples:

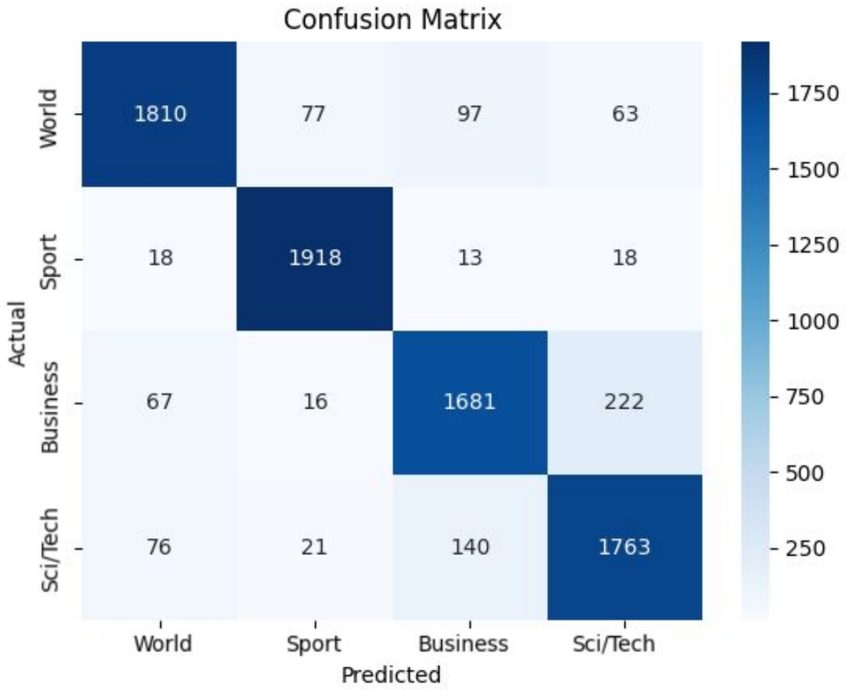
“Boeing fires airborne laser as part of missile defense A Boeing Co.-led team has succeeded in firing a laser beam for the first time as part of a ballistic missile defense shield, the Pentagon and the Boeing Co.”

“MessageLabs taps Brightmail in war on spam Email filtering firm MessageLabs yesterday announced a deal to incorporate Symantec's Brightmail anti-spam technology into its own anti-spam service.”



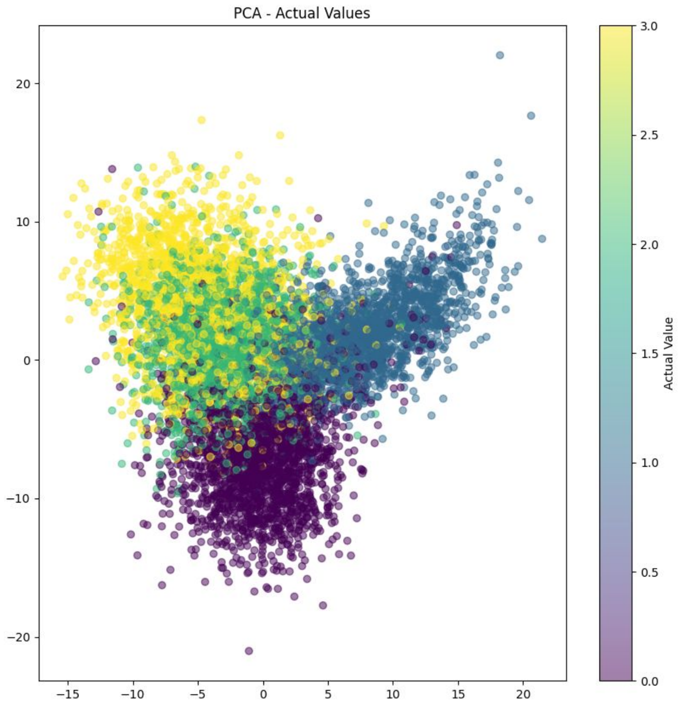
Accuracy: 89.675%, **F1**: 89.643%, **Precision**: 89.648%, **Recall**: 89.675%

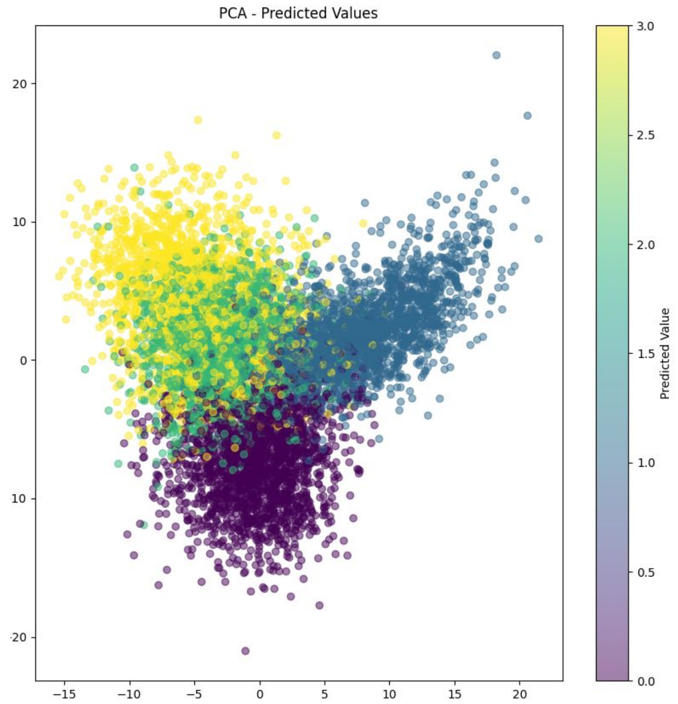
Support Vector Machine *Word Embeddings*



Accuracy: 89.65%, **F1**: 89.626%, **Precision**: 89.65%, **Recall**: 89.65%

Support Vector Machine *Word Embeddings*





Accuracy: 89.65%, **F1**: 89.626%, **Precision**: 89.65%, **Recall**: 89.65%

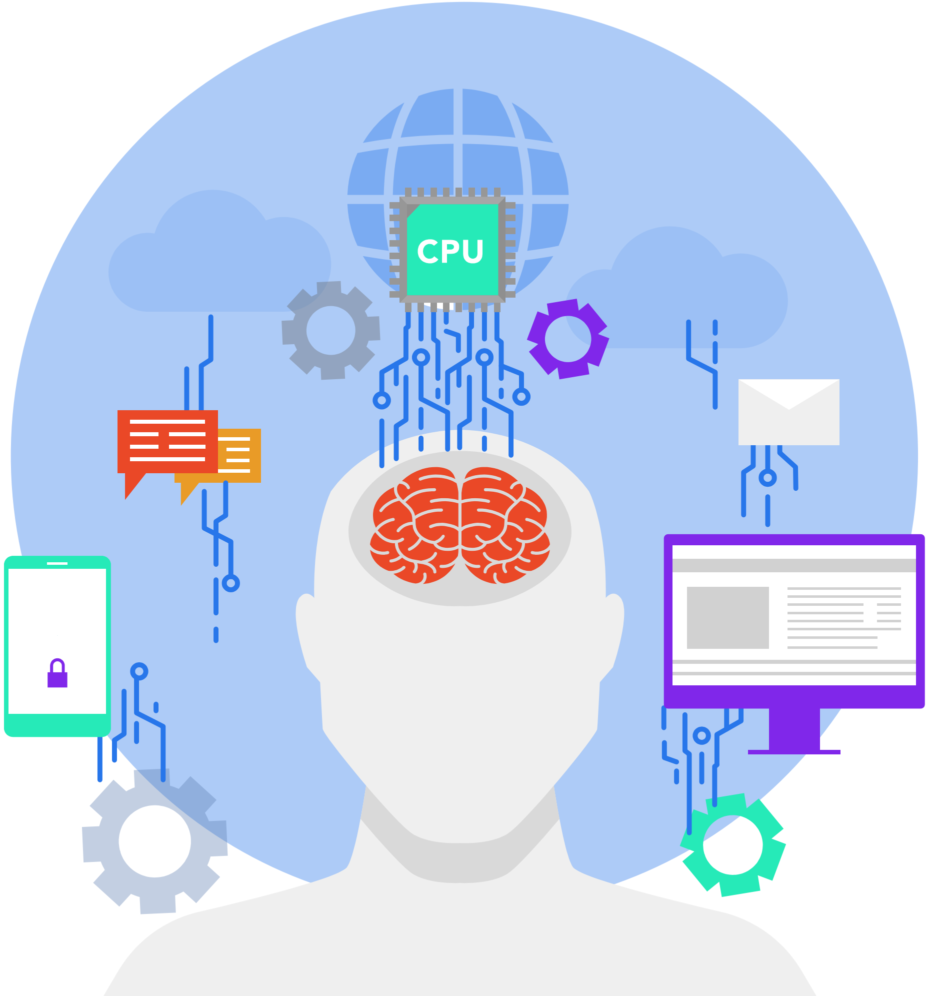
Conclusions

In the **World** category, both False Positives (FP) and True Negatives (TN) appear across all labels. This arises due to the wide breadth of topics and the presence of terms that may also relate to other labels, such as countries, nationalities, and companies. Consequently, texts on this topic have high ambiguity.

Conversely, the **Sports** category has high accuracy since its content is more specific, often mentioning distinct clubs, their respective countries, and the sports themselves. Hence, such specific texts are more easily classifiable.

However, the **Business** and **Sci/Tech** categories have their own challenges. These categories exhibit significant content overlap, particularly concerning entities like companies. Consequently, the classifier may struggle to differentiate between them, resulting in a notable number of both FP and TN.


Q&A



Thanks!

Bárbara Rodrigues
up202007163

Guilherme Pereira
up202007375

Lucas Sousa
up202004682