Welcome to the CoGrammar

Tutorial: Natural Language Processing

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Data Science Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Data Science Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Skills Bootcamp
8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

CoGrammar

Skills Bootcamp Progression Overview

✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar

Tutorial: Natural Language Processing

May 2024

Learning Objectives

Recap of **natural language processing**.

- **Text cleaning** using **RegEx**
- **spaCy pipeline**
- **Text preprocessing:** *tokenisation*, *stemming or lemmatisation*, *stop-word removal*, *parts-of-speech (POS) tagging*, and *named entity recognition (NER)*

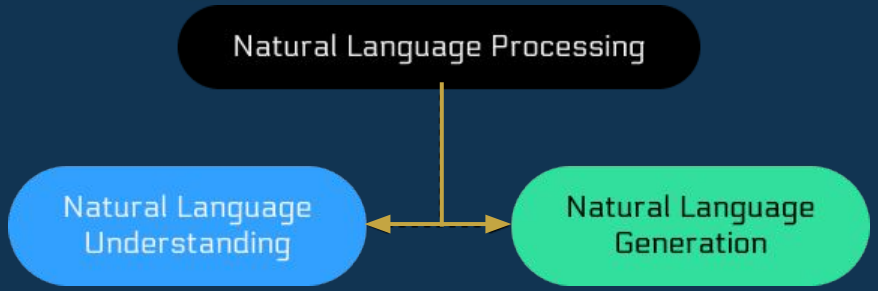
Learning Objectives

- **Feature engineering**: *spaCy similarity*, *Bag-of-Words*, *TF-IDF using sklearn*
- *Model building and evaluation:* **Sentiment analysis** and **text classification** using *spaCy* and *sklearn*

Recap of Natural Language Processing

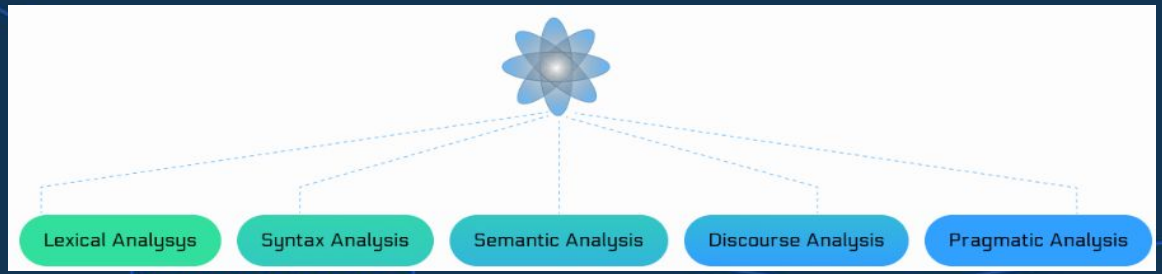
CoGrammar

NLP Components and Levels

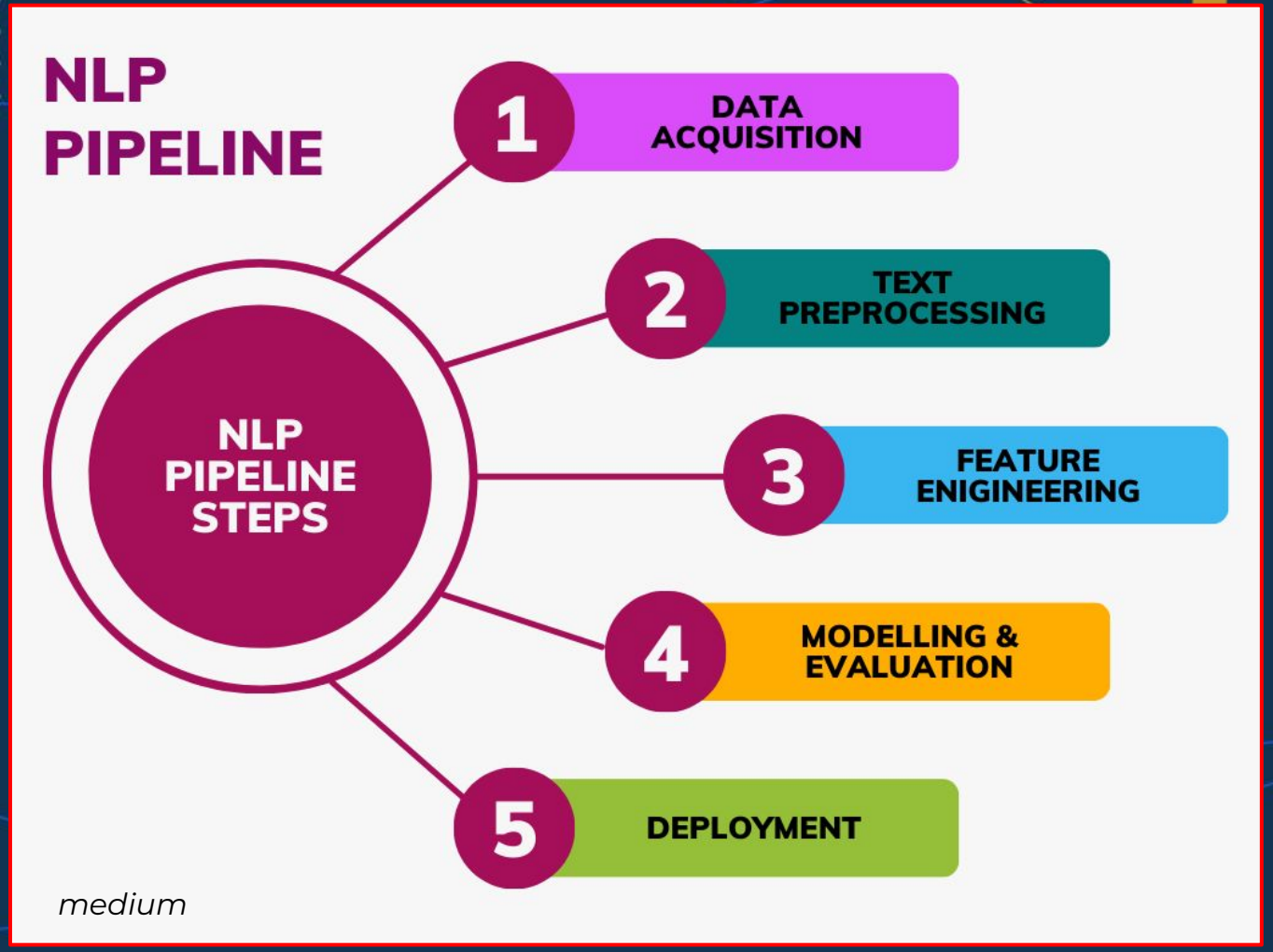


amazinum

- **Morphological/Lexical analysis:** processing and understanding POS.
- **Syntactic analysis:** understanding the sentence structure.
- **Semantic analysis:** understanding literal meaning of words, phrases, sentences.
- **Discourse analysis:** understanding units larger than single sentence
- **Pragmatic analysis:** using real-world knowledge to understand the bigger context of the sentence.



NLP Pipeline



Text Cleaning

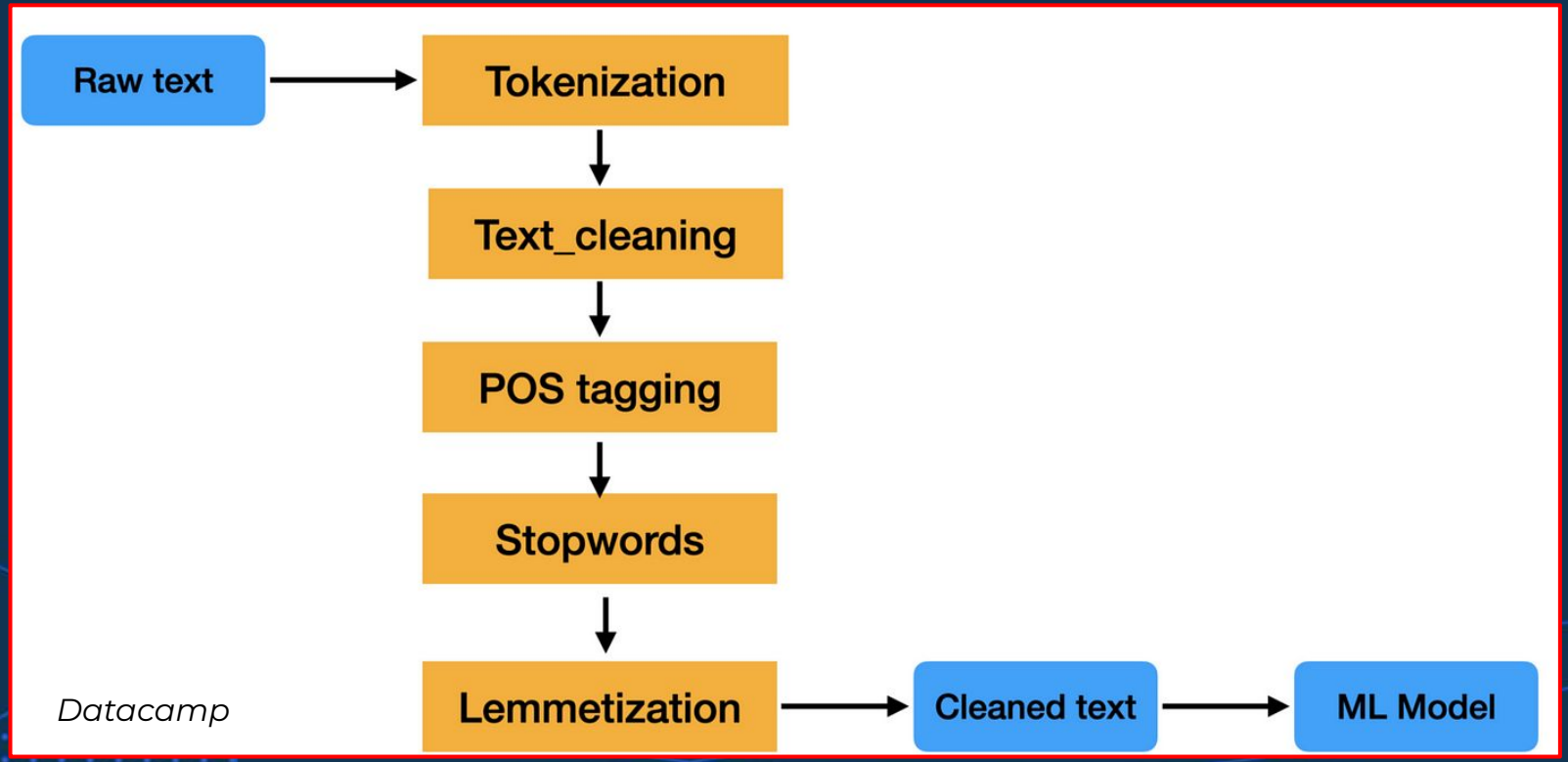
```
sample = ("Hello @billieeilish 👋🏾, still want us to \
          hit that new sushi spot 😡😡😡😡😡😡😡😡😡😡😡😡 ??? \
          LMK when you're free cuz I can't go this or next weekend \
          since I'll be swimming 🏊‍♀️!!! #cheerssushi 🍣 🐡")
```

Use regular expressions to remove unwanted characters.

```
'hello still want us to hit that new sushi spot lmk when youre free cuz i cant go this or next weekend since ill be swimming'
```

CoGrammar

Text Preprocessing



Text Preprocessing

- **Tokenisation:** segmenting text into list of tokens. For sentence tokenisation, tokens are sentences; for word tokenisation they are words.
- **Stemming** or **lemmatisation**: reduce words to their base form. Stemming involves stripping suffixes from words to get their stem. Lemmatisation involves reducing words to their base form based on their part of speech.
- **Stop word removal:** removal of commonly occurring words
- **POS tagging:** assign a part of speech tag to each word in a text.
- **Named Entity Recognition (NER):** identifying and classifying named entities in text, such as people, organisations, and locations.

Feature Engineering

- **Word embeddings** are dense vector representations of words, each word is represented as a high-dimensional vector in a continuous space.

- **Semantic similarity** is about the meaning closeness, and **lexical similarity** is about the closeness of the word set.

  - *"The dog bites the man"* and *"The man bites the dog"*
  - Identical considering lexical similarity; however entirely different considering semantic similarity

- **Cosine similarity** in NLP domain: measures the cosine of the angle between vectors of two points.

Semantic Similarity

- Use ***spacy*** pre-trained model with embeddings "***en_core_web_md***" and '***similarity***' to calculate the similarities between embeddings.
- **Bag-of-Words (BoW)** considers only word frequencies within a document and treats all words equally
- **Term Frequency-Inverse Document Frequency (TF-IDF)**: differentiates between common and rare words.
- Use ***CountVectorizer*** and ***TfidfVectorizer*** from ***sklearn*** to implement the Bag-of-Words and TF-IDF techniques.

Recap Q&A

CoGrammar

What is the primary challenge of NLP?

1. Handling ambiguity in natural language
2. Handling tokenisation
3. Handling POS tagging
4. All of the above

What is the primary challenge of NLP?

1. **Handling ambiguity in natural language**
2. Handling tokenisation
3. Handling POS tagging
4. All of the above

Parts-of-speech (POS) tagging determines

1. POS for each word dynamically as per meaning of the sentence
2. POS for each word dynamically as per sentence structure
3. All POS for a specific word given as input
4. All of the above

Parts-of-speech (POS) tagging determines

1. POS for each word dynamically as per meaning of the sentence
2. POS for each word dynamically as per sentence structure
3. All POS for a specific word given as input
4. **All of the above**

CoGrammar

The process of understanding the meaning and interpretation of words, signs, and sentence structure is called as

1. Tokenisation
2. Lexical Analysis
3. Semantic Analysis
4. Sentiment Analysis

The process of understanding the meaning and interpretation of words, signs, and sentence structure is called as

1. Tokenisation
2. Lexical Analysis
3. **Semantic Analysis**
4. Sentiment Analysis

The sentence "I saw bats" contains which type of ambiguity?

1. Syntactic
2. Semantic
3. Lexical
4. Pragmatic

The sentence "I saw bats" contains which type of ambiguity?

1. Syntactic
2. Semantic
3. **Lexical**
4. Pragmatic

Linear sequences of words are transformed into structure that show how the words are related to each other is

1. Syntactic analysis
2. Semantic analysis
3. Lexical analysis
4. Pragmatic analysis

Linear sequences of words are transformed into structure that show how the words are related to each other is

1. **Syntactic analysis**
2. Semantic analysis
3. Lexical analysis
4. Pragmatic analysis

CoGrammar

Model Building and Evaluation

Sentiment Analysis

CoGrammar

Dataset

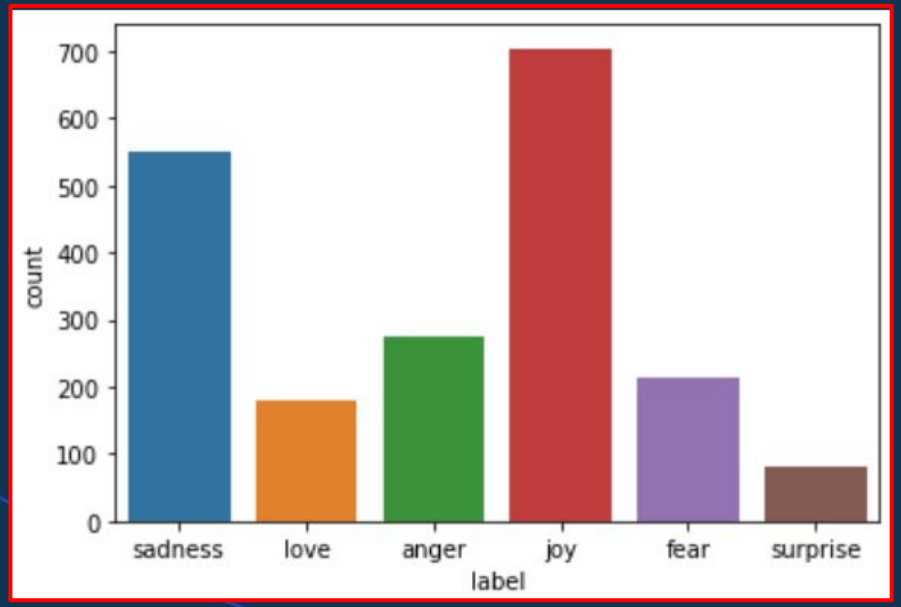
Training and Test Dataset from Kaggle
(originally from https://aclanthology.org/D18-1404/)

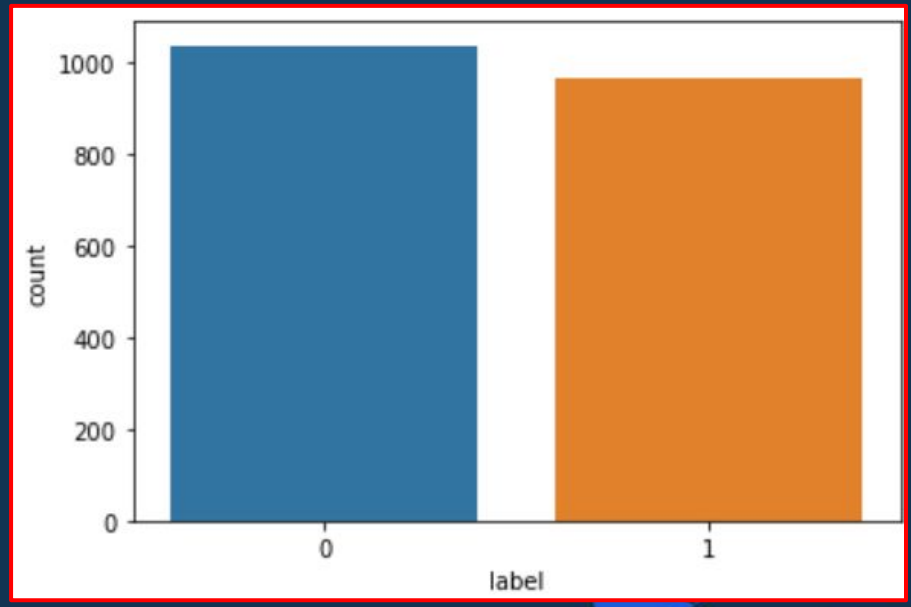
```
im feeling quite sad and sorry for myself but ill snap out of it soon;sadness
i feel like i am still looking at a blank canvas blank pieces of paper;sadness
i feel like a faithful servant;love
i am just feeling cranky and blue;anger
i can have for a treat or if i am feeling festive;joy
i start to feel more appreciative of what god has done for me;joy
i am feeling more confident that we will be able to take care of this baby;joy
i feel incredibly lucky just to be able to talk to her;joy
i feel less keen about the army every day;joy
i feel dirty and ashamed for saying that;sadness
i feel bitchy but not defeated yet;anger
i was dribbling on mums coffee table looking out of the window and feeling very happy;joy
i woke up often got up around am feeling pukey radiation and groggy;sadness
```

CoGrammar

We will use a subset of the training dataset (2,000 out of 18,000)

Dataset





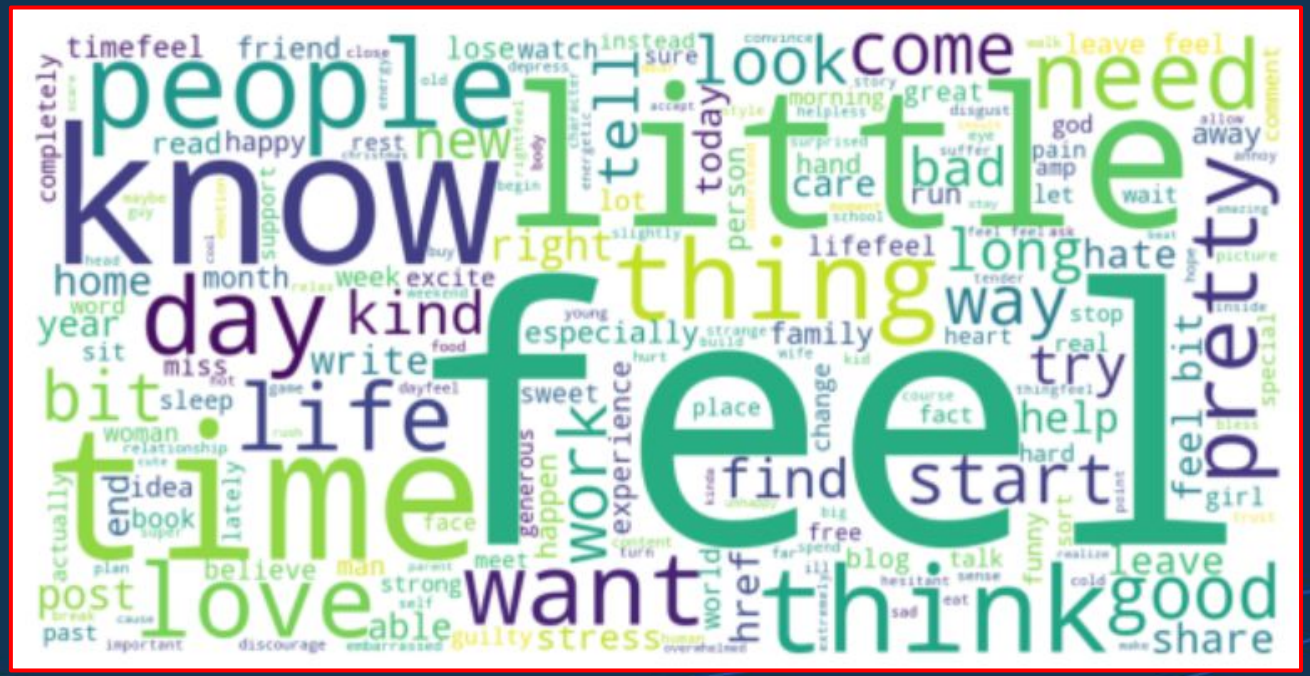
Positive Sentiment – `joy`, `love`, `surprise` -> 1

Negative Sentiment – `anger`, `sadness`, `fear` -> 0

CoGrammar

Wordcloud

Data visualization depicts the more frequent words appear enlarged as compared to less frequent words. This gives us a little insight into, how the data looks after being processed through all the steps.



CoGrammar

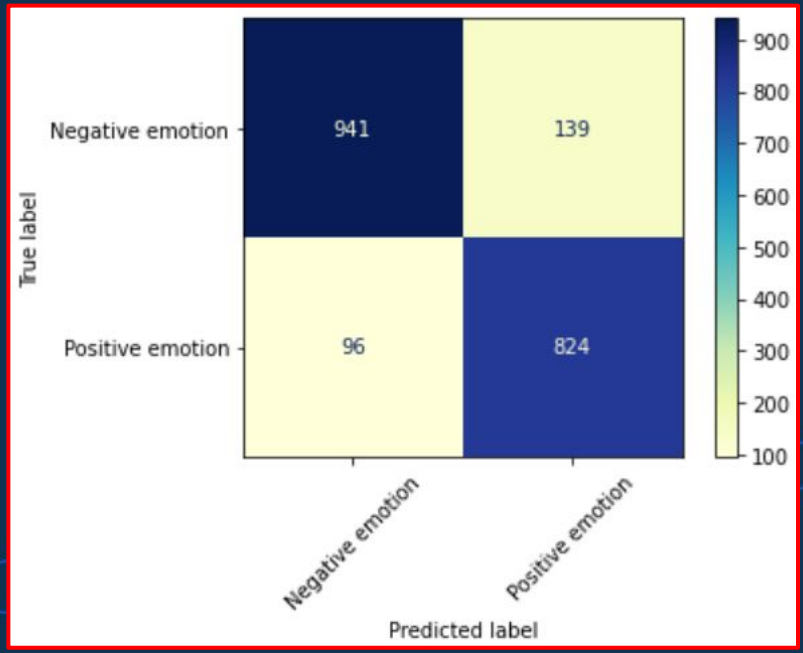
Random Forest Classifier

```
RandomForestClassifier
RandomForestClassifier(max_features=2, min_samples_split=5, n_estimators=500)
```

```
Accuracy_score:  0.8825
Precision_score:  0.8556593977154725
Recall_score:  0.8956521739130435
--------------------------------------------------
              precision    recall  f1-score   support

           0       0.91      0.87      0.89      1080
           1       0.86      0.90      0.88       920

    accuracy                           0.88      2000
   macro avg       0.88      0.88      0.88      2000
weighted avg       0.88      0.88      0.88      2000
```



CoGrammar

Model Building and Evaluation

Text classification

CoGrammar

Dataset

20 Newsgroups data set

- Originally collected by Ken Lang, “Newsweeder: Learning to filter netnews. http://qwone.com/~jason/20Newsgroups/
- Collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.
- Popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.

Loading the dataset

```
from sklearn.datasets import fetch_20newsgroups
twenty_train_all = fetch_20newsgroups(subset='train', shuffle=True, random_state=42)
twenty_train_all.target_names
```

The dataset has a total of 20 target names
We will work with a partial dataset with only 4 categories.

```
categories = ['comp.graphics', 'sci.space', 'sci.med','sci.crypt']
```

Feature extraction

Bag of Words

```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(twenty_train.data)
```

TF-IDF vectoriser

```
from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
```

Classification: Training and predictions

```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB

log_reg_clf = LogisticRegression().fit(X_train_tfidf, twenty_train.target)
nb_clf = MultinomialNB().fit(X_train_tfidf, twenty_train.target)
```

```
docs_new = ['Andromeda galaxy is nearest to the solar system', 'OpenGL on the GPU is fast']
```

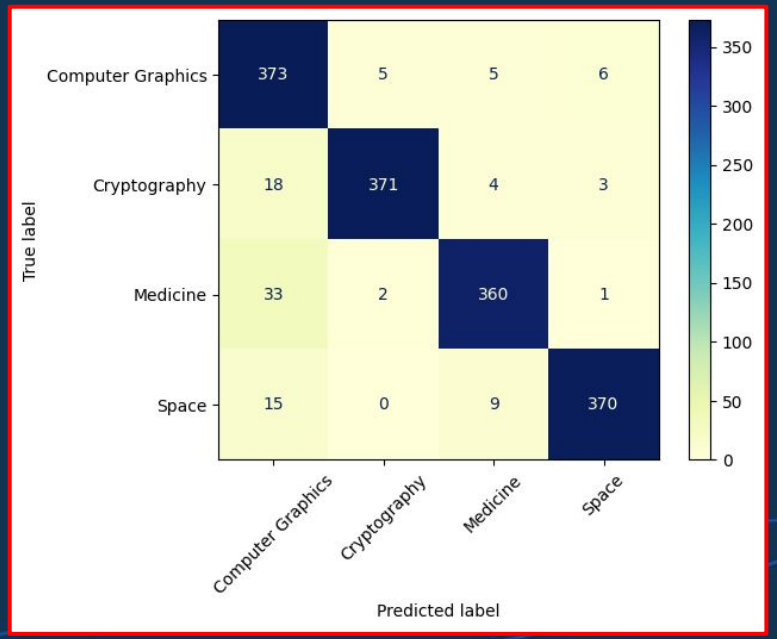
Predictions

```
Logistic Regression
'Andromeda galaxy is nearest to the solar system' => sci.space
'OpenGL on the GPU is fast' => comp.graphics
Naive Bayes
'Andromeda galaxy is nearest to the solar system' => sci.space
'OpenGL on the GPU is fast' => comp.graphics
```

Model Evaluation

Logistic Regression Accuracy: **91.4%**
SGDClassifier Accuracy: **94%**

	precision	recall	f1-score	support
comp.graphics	0.85	0.96	0.90	389
sci.crypt	0.98	0.94	0.96	396
sci.med	0.95	0.91	0.93	396
sci.space	0.97	0.94	0.96	394
accuracy			0.94	1575
macro avg	0.94	0.94	0.94	1575
weighted avg	0.94	0.94	0.94	1575



CoGrammar

Key points

- NLP needs extra processing steps compared to general machine learning pipelines as there are added challenges to natural language e.g. text data.
- **Text cleaning:** essential to prepare for NLP tasks. **Regular Expression** is used for searching strings of specific patterns to convert or remove them.
- **Text Preprocessing** includes **tokenisation, stemming** or **lemmatisation**, **stop-word removal, parts-of-speech tagging** and **named entity recognition.**
- **Feature Engineering:** represent text in numeric vectors for the ML algorithm to understand the text attribute.
- **Model Building and Evaluation**

Limitations of NLP

- **Language differences, multilingualism:** challenges in understanding of natural language, more for rare languages.
- **Training data: vast** and **diverse** training data needed with **high-quality annotations**.
- **Time and Resource Requirements**
  - Time consuming and resource intensive to collect, annotate, and preprocess the large text datasets.
  - Powerful computation resources and time for training algorithms.
- **Mitigating innate Biases in NLP Algorithms**

Further resources

- https://www.deeplearning.ai/resources/natural-language-processing/
- https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline/
- https://spacy.io/models
- https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
- https://www.analyticsvidhya.com/blog/2021/05/natural-language-processing-step-by-step-guide/

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar