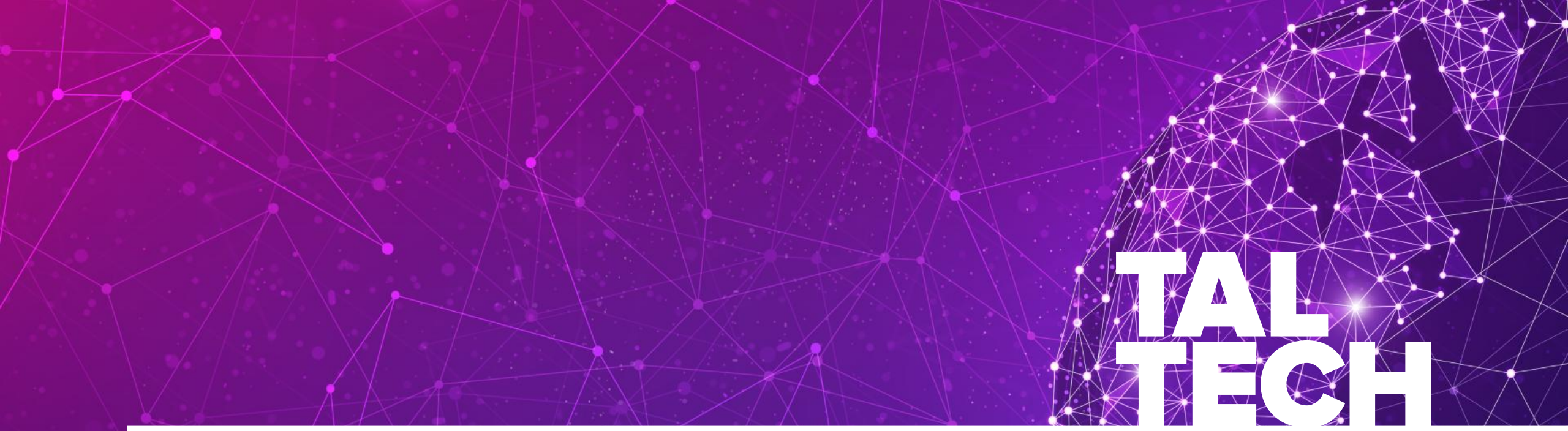


Application of Machine Learning for HS-6 Code Assignment

Deniss Ruder

18.08.2020

MOTIVATION

The European Commission ”e-commerce package” coming into force on the **1st of July 2021**:

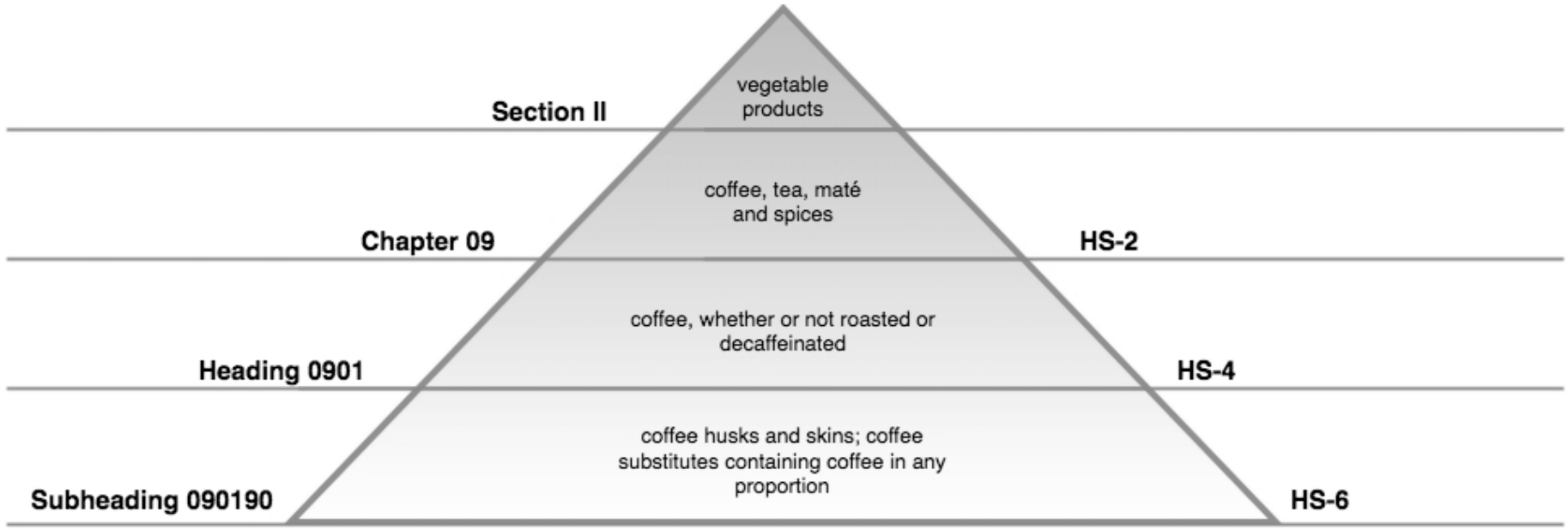
- Rejection of VAT exemption
- Extension of ”Mini One Stop Shop” to ”**One Stop Shop**” (**OSS**)

The purpose of OSS is to solve two main problems that cause tax losses:

- **Fraud** attempts
- **Manual Harmonized System** (**HS**) **code assignment** by sellers, senders, shippers, and carriers

HARMONIZED SYSTEM CODE

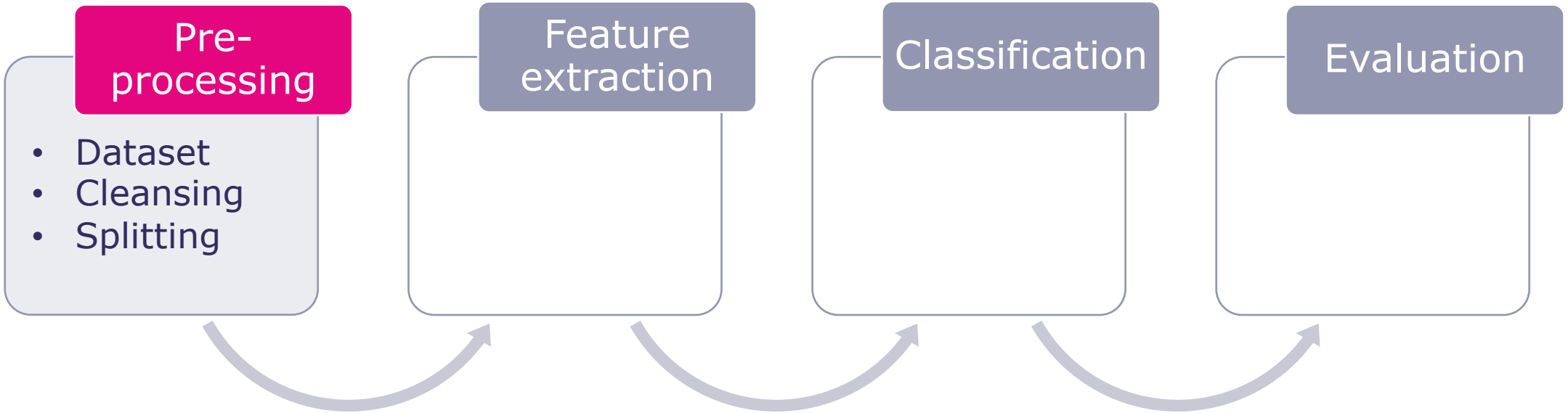
The basic **HS-6** (required in the EU) includes 21 sections, 99 chapters, 1244 headings, and 5224 subheadings. For instance, **090190** stands for:



DATABASE SEARCH

Database queries do not give any meaningful results for searching HS codes according to cargo descriptions due to:

- **The HS complexity and hard to follow guidelines**
- **The HS continuous revisions**
- **The cargo descriptions noisiness**
- **The gap in terminology**: For instance, ”Apple Ipad” could be classified as **847141** - ”Automatic data-processing machines and units thereof; magnetic or optical readers, machines for transcribing data onto data media in coded form and machines for processing such data, not elsewhere specified or included”.


Pre-processing
• Dataset
• Cleansing
• Splitting
Feature extraction
Classification
Evaluation

DATASET

US Imports 2018-2020 is a dataset weekly gathered by the US Customs and Border Protection agency and publicly available from amazon AWS.

Besides other fields, the dataset contains tables with HS code and cargo descriptions columns. Initially, these tables held **124,000,000** and **41,000,000** entries.

Cargo description	HS code
WOODWORKING MACHINE AND SPARE PARTS WEIGHT 8...	846591
WOODWORKING MACHINE AND SPARE PARTS PO NO. 751...	846591
STAND, ZERO CLEARANCE THROAT PLATE, GLIDE PAD) ...	846591
...	...

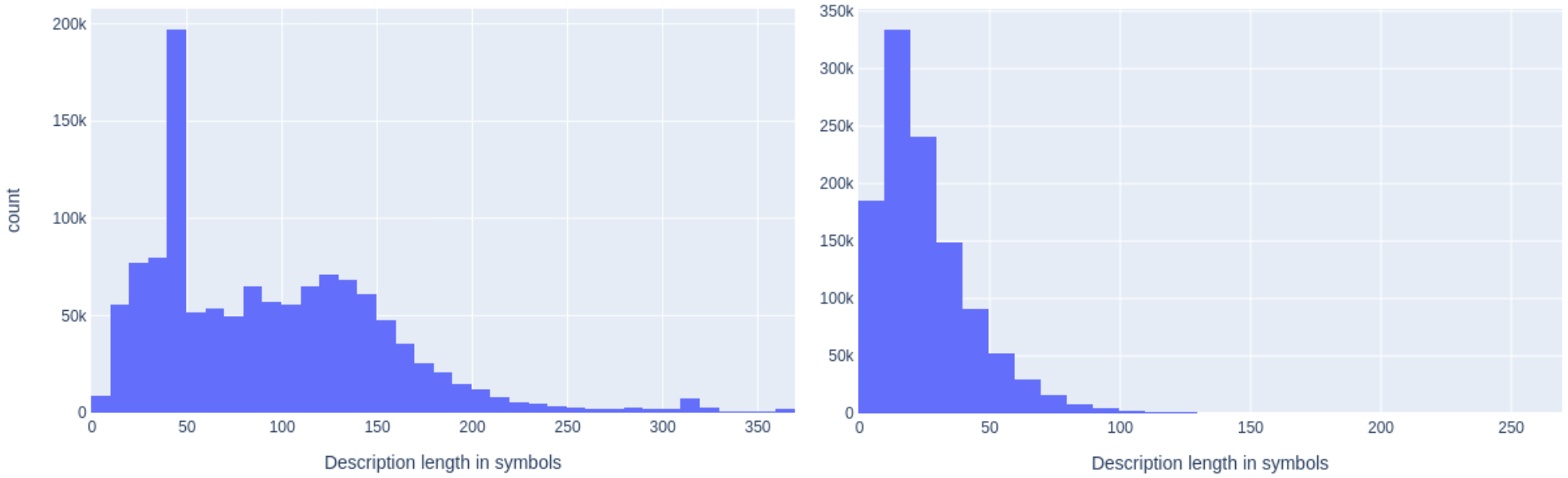
CLEANSING

The data pre-processing included 19 separate steps, where most impactful were **the normalization** procedures:

- Converting to lower case
- Removing punctuation and stop words
- Removing words with non-alphabetic characters
- Lemmatization
- Part-of-speech (POS) tagging
- Removing non-English words
- Noise removal

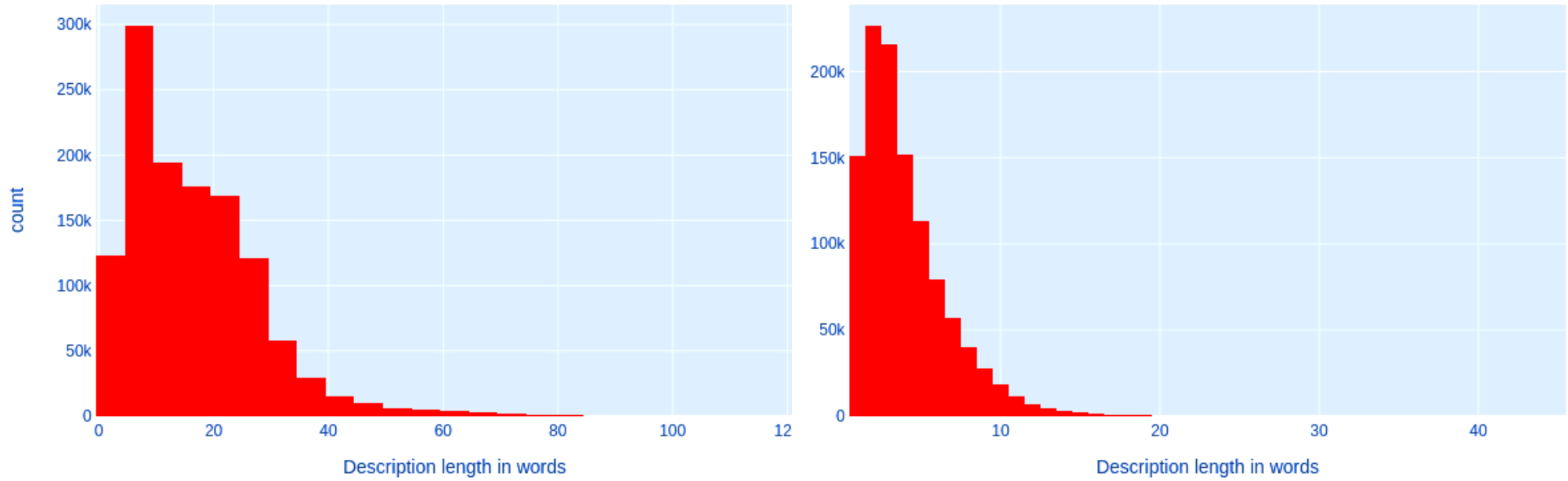
CLEANSING

The average description length in symbols reduced from **94.11** to **25.33**.



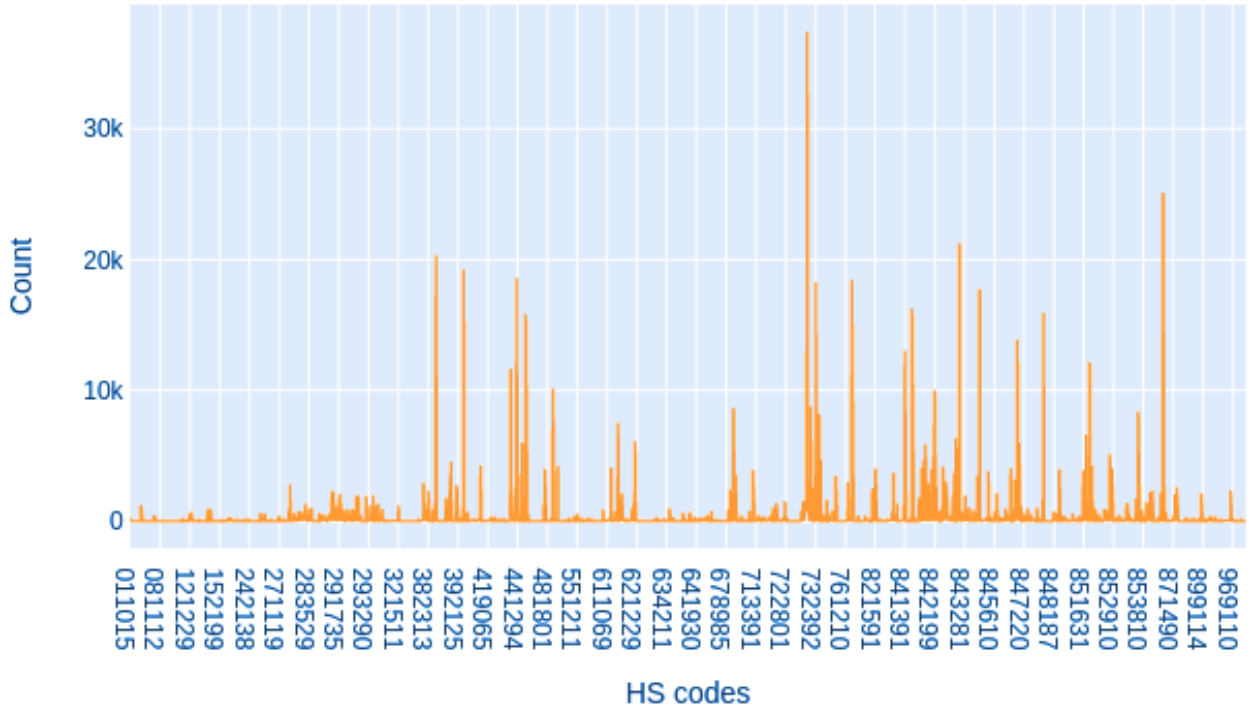
CLEANSING

The average description length in words reduced from **16.60** to **4.01**.



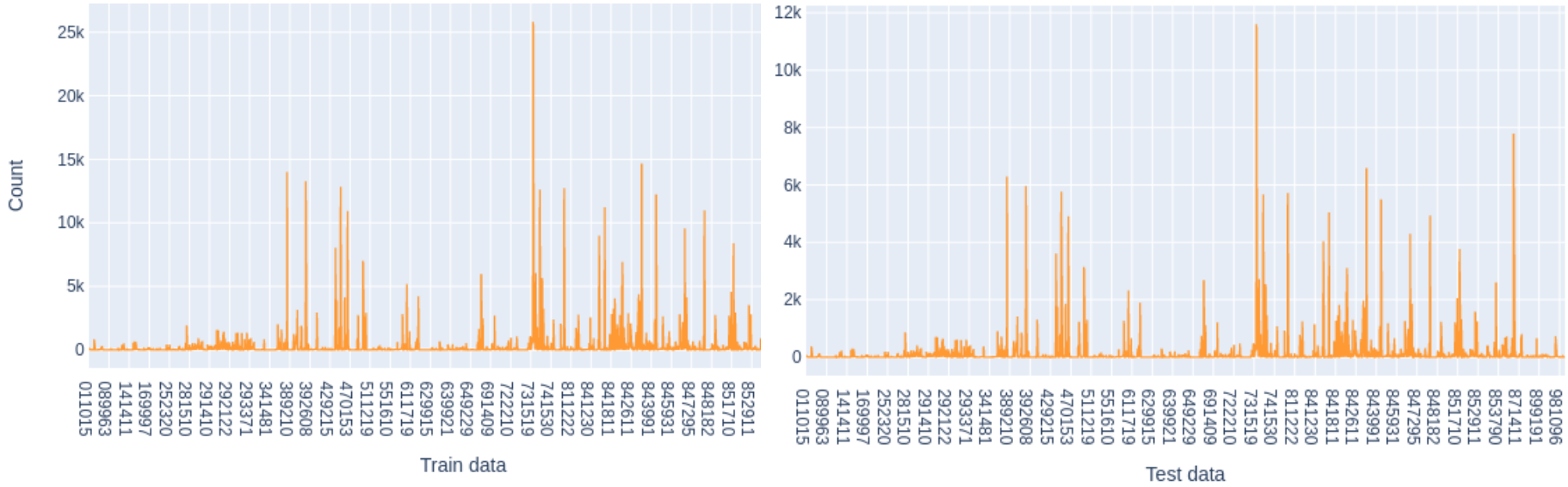
CLEANSING

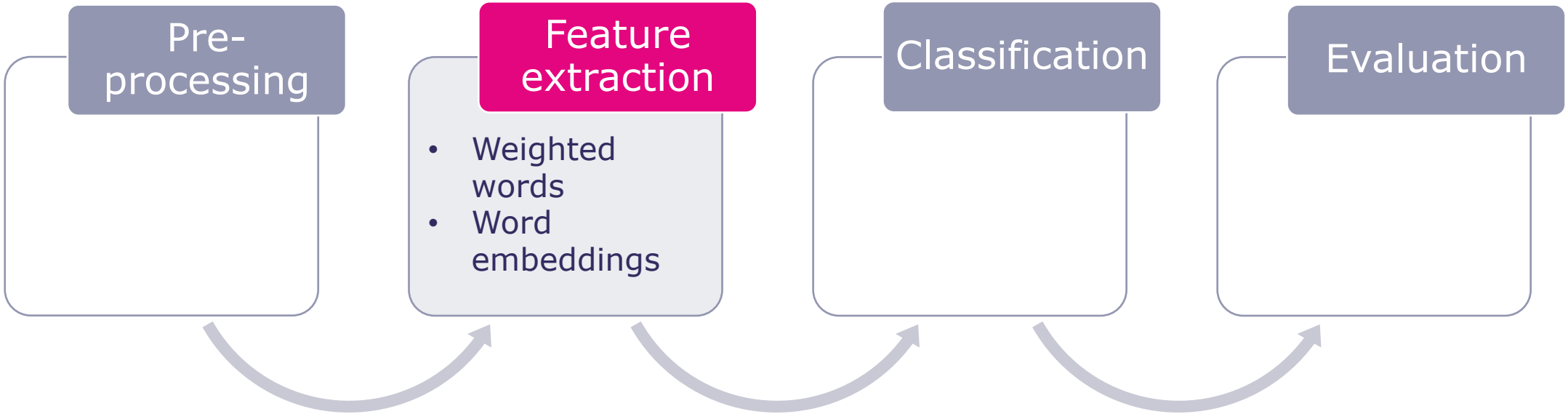
After the pre-processing steps, the dataset remained **1,124,874** records with **17,476** unique words, and **3243** unevenly distributed HS codes:



SPLITTING

To keep all class occurrences in both train and test datasets, we split it in the proportion of **0.69/0.31**, applying **stratified sampling**.




Pre-processing
Feature extraction
• Weighted words
• Word embeddings
Classification
Evaluation

WEIGHTED WORDS

Most applied text classifiers use the **TF-IDF** weighting scheme:

”This is a funny example”	[”funny”, ”example”]
”This is a very sad example”	[”very”, ”sad”, ”example”]

TF = (Number of repetitions of word in a document) / (Number of words in a document)

IDF = Log[(Number of documents) / (Number of documents containing the word)]

$$\textbf{TF-IDF} = \text{tf}(\text{“funny”}) * \text{idf}(\text{“funny”}) = \frac{1}{2} * \log\left(\frac{1}{2}\right) \approx -0.1505$$

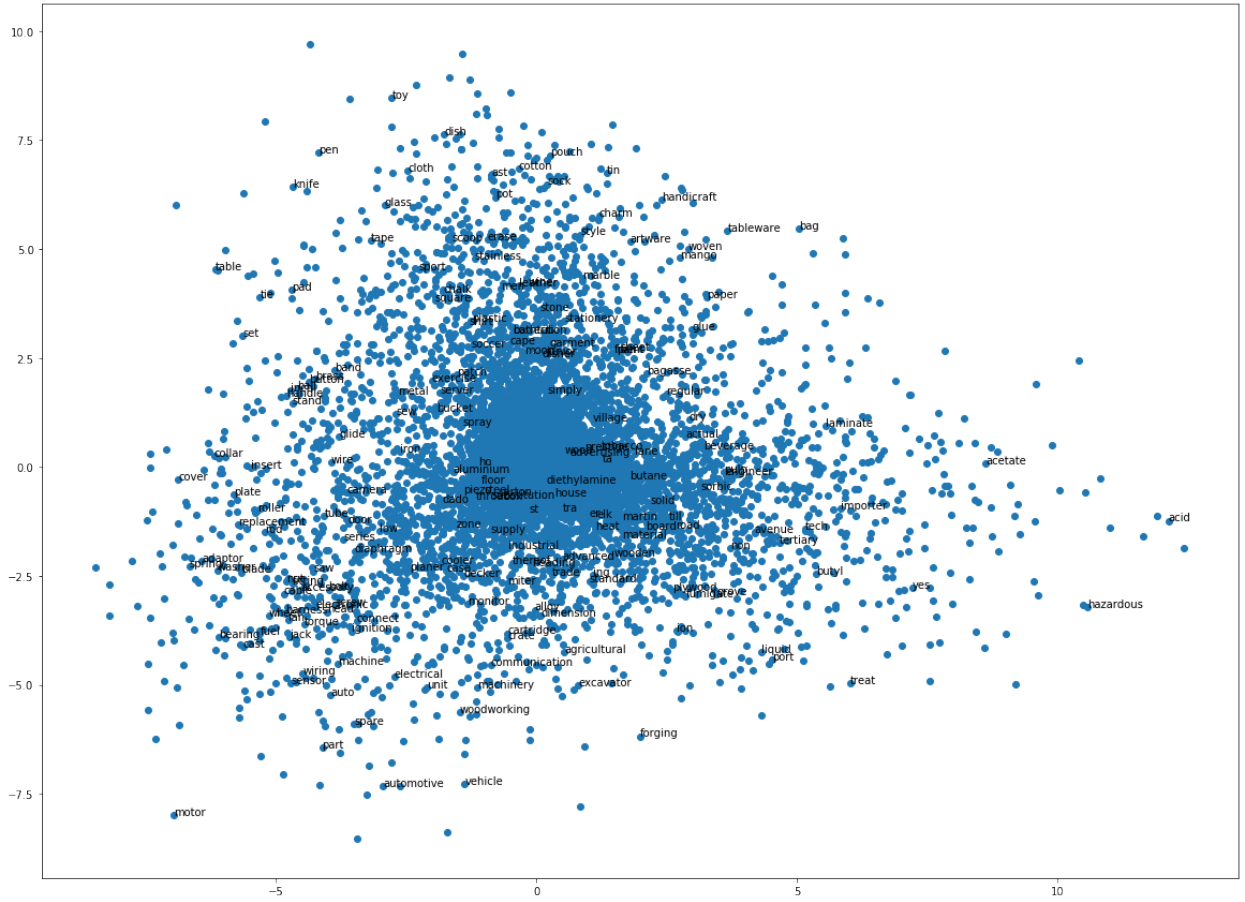
$$\text{Vocabulary} = \{\ \text{”funny”},\ \text{“very”},\ \text{”sad”},\ \text{”example”}\}$$

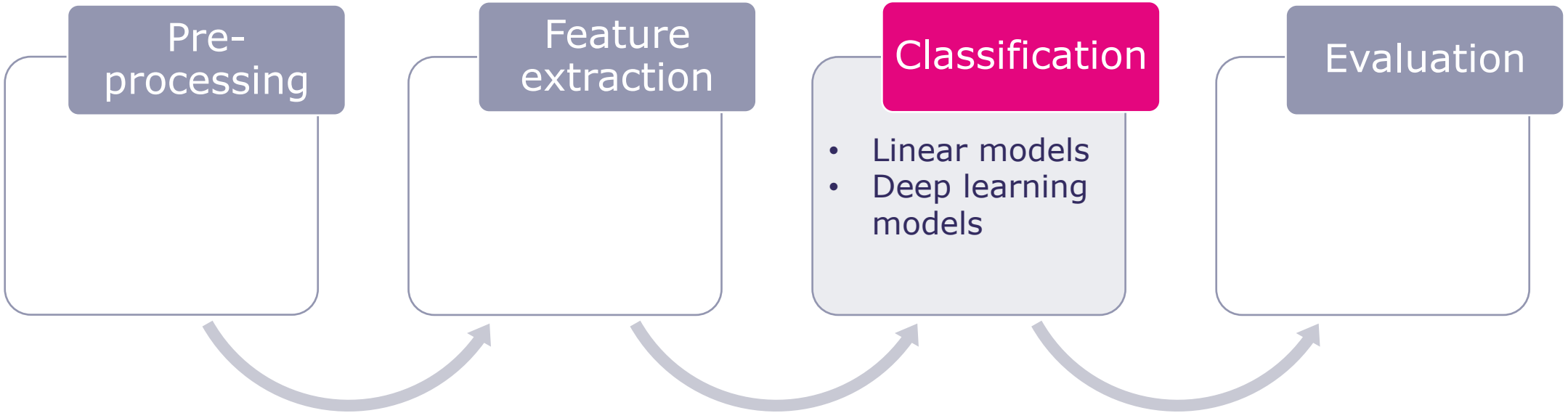
$$\text{Resulting feature matrix} = \begin{bmatrix} -0.1505 & 0. & 0. & 0. \\ 0. & -0.1003 & -0.1003 & 0. \end{bmatrix}$$

WORD EMBEDDINGS

Used word embedding models:

- Word2Vec Skip-gram
- Word2Vec CBOW
- Doc2Vec PV-DBOW
- Doc2Vec PV-DM
- GloVe




Pre-processing
Feature extraction
Classification
• Linear models
• Deep learning models
Evaluation

LINEAR MODELS

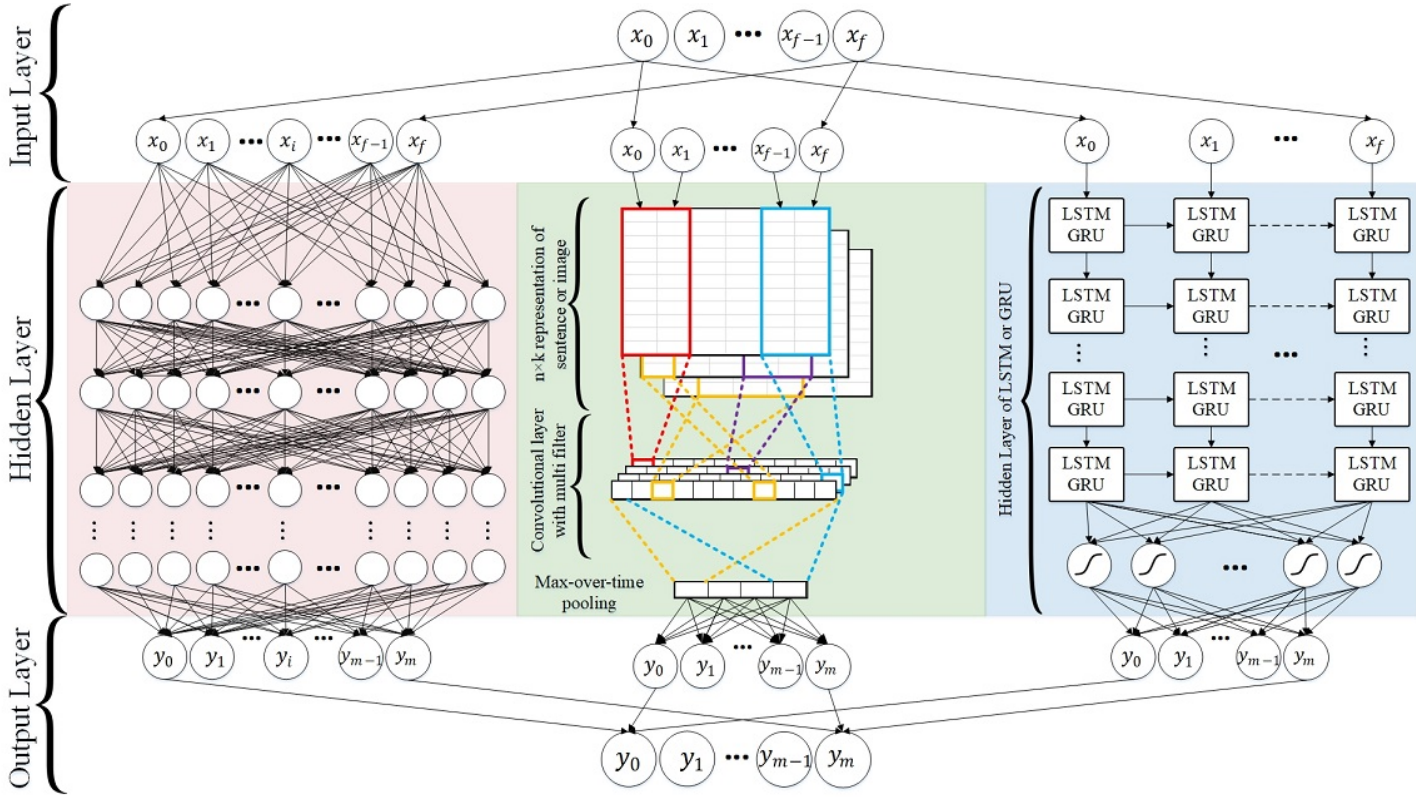
Tested linear classification models that use TF-IDF feature vectors as input:

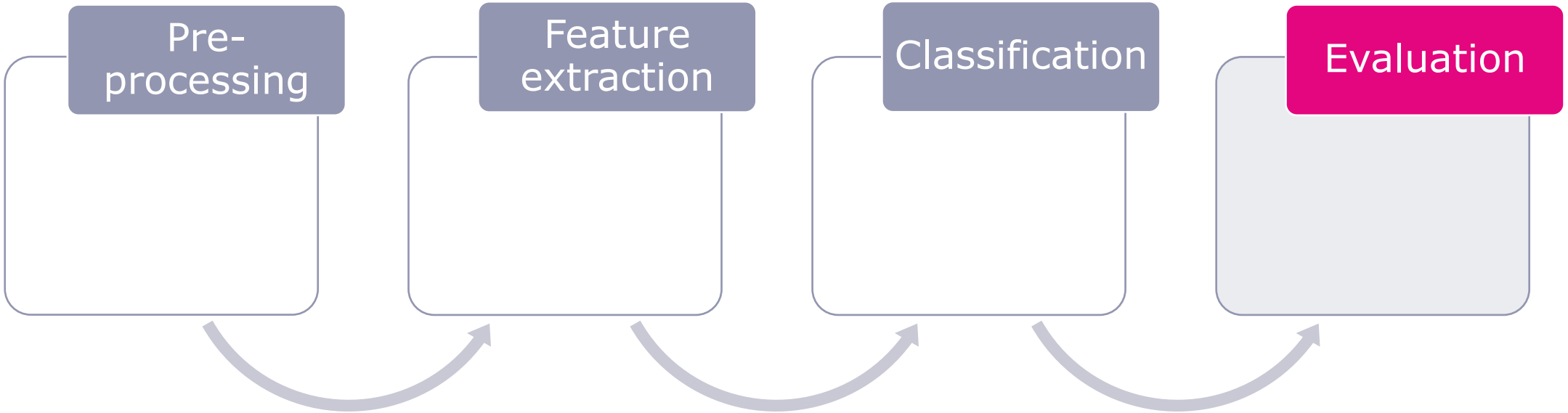
- Rocchio classification
- Multinomial Logistic Regression (MLR)
- Multinomial Naïve Bayes (MNB)
- K-Nearest Neighbor (k-NN)
- Decision tree
- Random forest
- Support Vector Machine (SVM)

DEEP LEARNING MODELS

Deep learning models found by (RDML):

- **DNN** (TF-IDF)
- **CNN** (Word embeddings)




Pre-
processing
Feature
extraction
Classification
Evaluation

EVALUATION

The metrics we orient when evaluating our classification models:

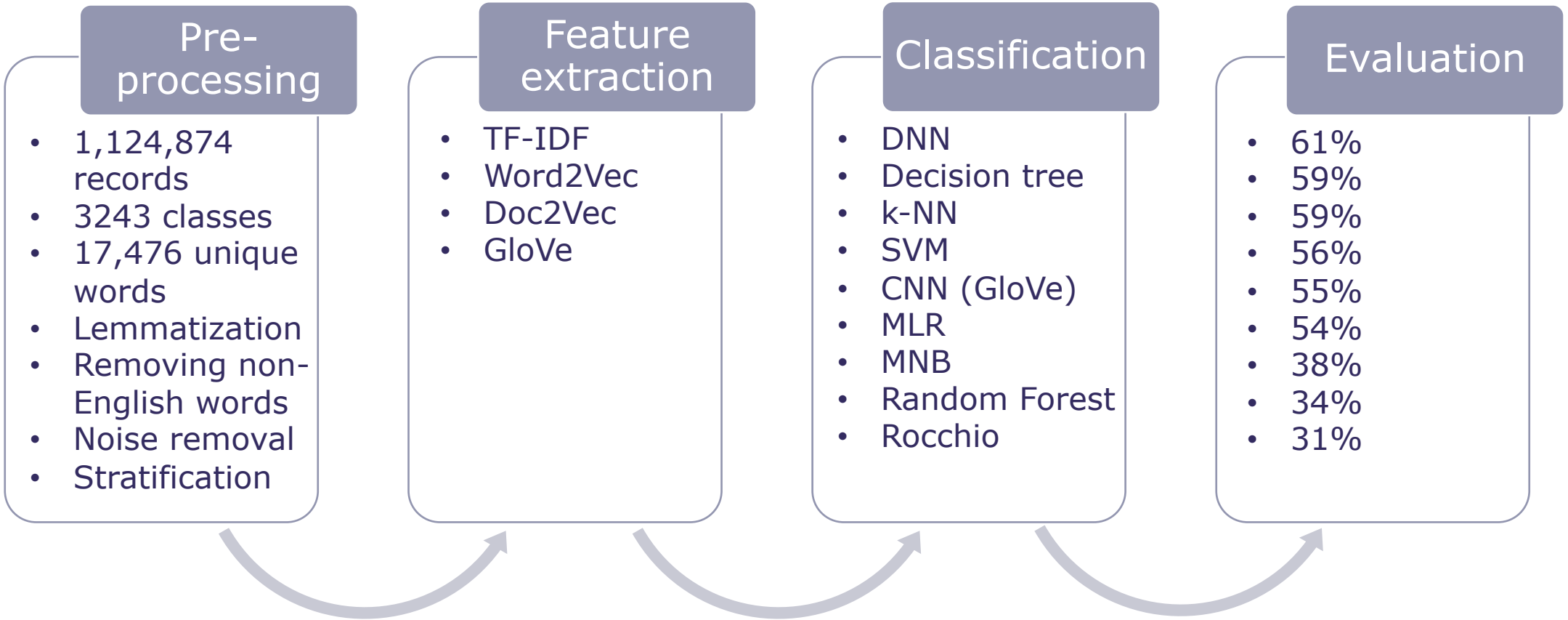
- Confusion matrix
- Support
- F-1 accuracy
- F-1 macro average
- **F-1 weighted average**

F-1 Weighted average considers the performance of every individual class, which is relevant for our unbalanced dataset.

RESULTS

Classifier	Features	F-1 macro avg	F-1 accuracy	F-1 weighted avg
DNN	**TF-IDF**	**0.25**	**0.62**	**0.61**
Decision tree	TF-IDF	0.30	0.60	0.59
k-NN	TF-IDF	0.27	0.59	0.59
SVM	TF-IDF	0.28	0.58	0.56
CNN	GloVe	0.15	0.57	0.55
CNN	Word2Vec Skip-gram	0.15	0.57	0.55
MLR	TF-IDF	0.18	0.56	0.54
CNN	Word2Vec CBOW	0.15	0.56	0.54
CNN	Doc2Vec PV-DBOW	0.15	0.56	0.54
CNN	Doc2Vec PV-DM	0.14	0.56	0.53
MNB	TF-IDF	0.05	0.43	0.38
Random Forest	TF-IDF	0.05	0.35	0.34
Rocchio	TF-IDF	0.16	0.26	0.31

SUMMARY


Pre-processing
1,124,874 records
3243 classes
17,476 unique words
Lemmatization
Removing non-English words
Noise removal
Stratification
Feature extraction
TF-IDF
Word2Vec
Doc2Vec
GloVe
Classification
DNN
Decision tree
k-NN
SVM
CNN (GloVe)
MLR
MNB
Random Forest
Rocchio
Evaluation
61%
59%
59%
56%
55%
54%
38%
34%
31%