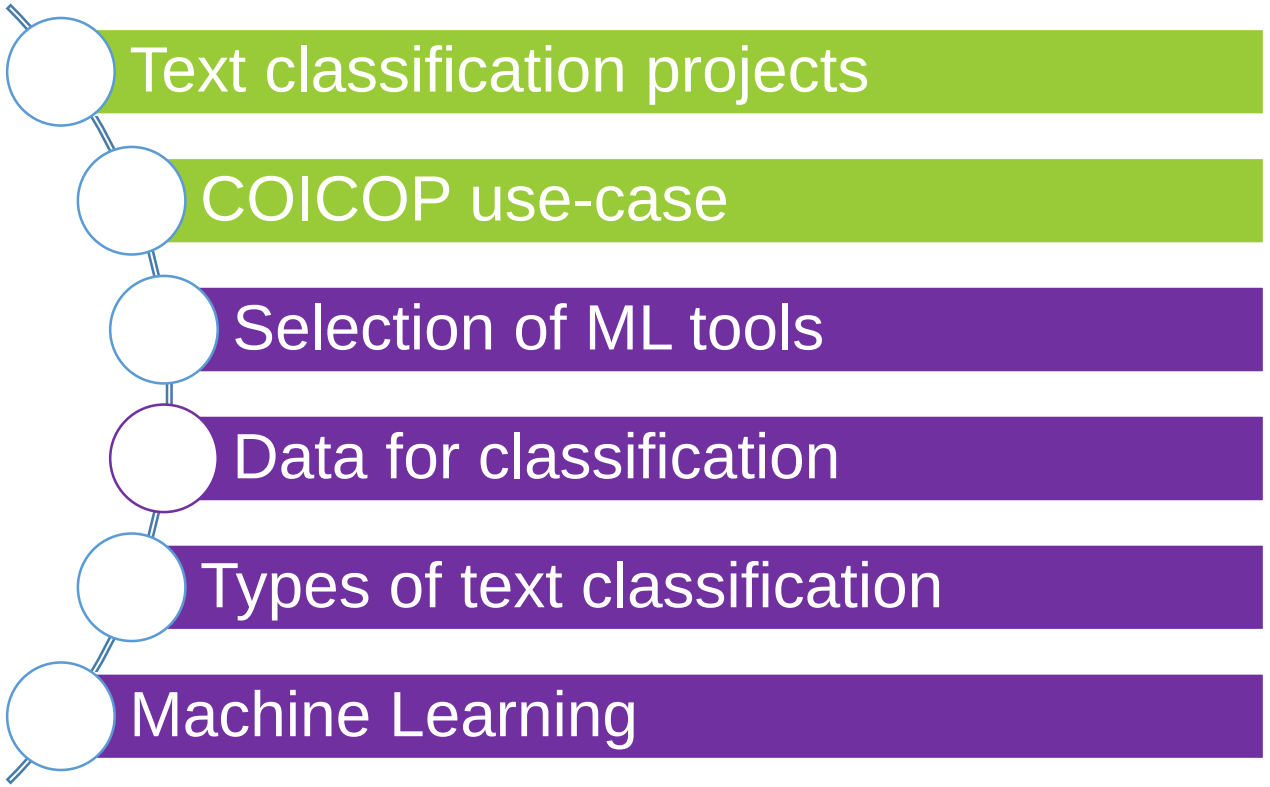
Statistics Poland

Text classification on webscraped data

Statistics Poland’s projects

Marcin Związek, Tomasz Markuszewski, Klaudia Peszat

Agenda


Text classification projects
COICOP use-case
Selection of ML tools
Data for classification
Types of text classification
Machine Learning

Text classification projects

1. OJA data → ISCO / ESCO classification
   - *ESSnet Big Data I and II, ESSnet Web Intelligence Network projects*
   - *https://github.com/WebIntelligenceNetwork/Deliverables/blob/main/WP4/D4_8.pdf*
   - *Experimental statistics*
2. Pharmaceutical and medical items → COICOP classification
   - *In production*
3. Computer articles → COICOP classification
4. Online real estate offers → is apartment furnished or not?
   - *Ongoing project*

COICOP use-case

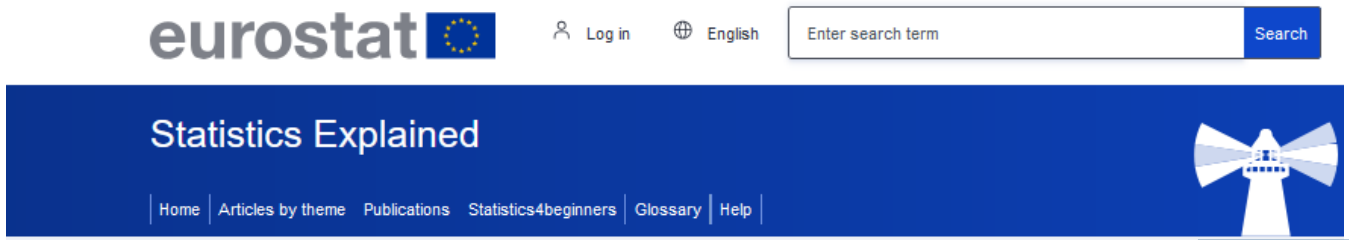
Automatic COICOP classification based on webscraping product names.

The purpose of the classification is to automate the process of assigning COICOP categories on data collected from online pharmacies.

Issues to be resolved:

- Infrastructure and selection of tools for text classification
- Data acquisition – source selection and assessment
- Preparing the text
- Selection and use ML text classifictaion algorithm
- Providing the ability to view data and correct the COICOP classification result for other people (via the Web Application)

COICOP classification



Main > COICOP HICP

Glossary:COICOP HICP

The **COICOP/HICP** is the United Nations (UN) Classification of individual consumption by purpose (COICOP), which was adapted to the compilation of the harmonised index of consumer prices (HICP) of the European Union (EU) and the euro area.

Adapting COICOP to the HICP calculation involved a number of changes:

- some sub-indices of the COICOP, such as narcotics and owner-occupied housing, had to be excluded because they are not within the HICP coverage;
- certain sub-classes (those with 4 digits) have been combined to ensure their weight was above one part per thousand in most of the Member States.

Example

COICOP 01-12 - Individual consumption expenditure of households

01 - FOOD AND NON-ALCOHOLIC BEVERAGES
- 01.1 - Food
- 01.2 - Non-alcoholic beverages

02 - ALCOHOLIC BEVERAGES AND TOBACCO
- 02.1 - Alcoholic beverages
- 02.2 - Tobacco

03 - CLOTHING AND FOOTWEAR
- 03.1 - Clothing
- 03.2 - Footwear

04 - HOUSING, WATER, GAS, ELECTRICITY AND OTHER FUELS
- 04.1 - Actual rentals for housing
- 04.3 - Regular maintenance and repair of the dwelling
- 04.4 - Other services relating to the dwelling
- 04.5 - Electricity, gas and other fuels

05 - FURNISHINGS, HOUSEHOLD EQUIPMENT AND ROUTINE MAINTENANCE OF THE HOUSE
- 05.1 - Furniture, furnishings and decorations, carpets and other floor coverings and repairs
- 05.2 - Household textiles
- 05.3 - Household appliances
- 05.4 - Glassware, tableware and household utensils
- 05.5 - Tools and equipment for house and garden
- 05.6 - Goods and services for routine household maintenance

06 - HEALTH
- 06.1 - Medical products, appliances and equipment
- 06.2 - Outpatient services
- 06.3 - Hospital services

1.Division (01)

2.Group (01.1)

3.Class (01.1.4)

4.Subclass (01.1.4.7)

ML tools selection

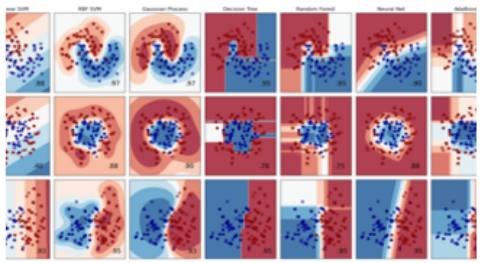
Which model?
Naive Bayes,SVM,
Logistic Regession,
Radom Forest

scikit learn Install User Guide API Examples Community More 1.5.2 (stable)

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

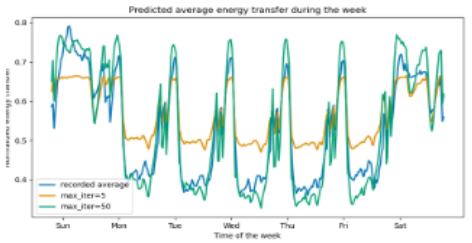


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...



Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...

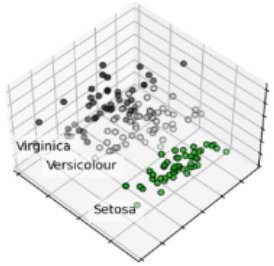
Examples

How ?

Dimensionality reduction

Reducing the number of random variables to consider.

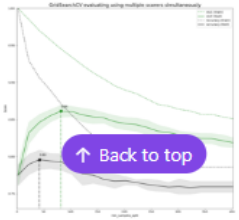
Applications: Visualization, increased efficiency.
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...



Model selection

Comparing, validating and choosing parameters and models.

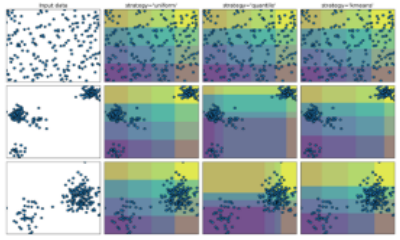
Applications: Improved accuracy via parameter tuning.
Algorithms: Grid search, cross validation, metrics, and



Preprocessing

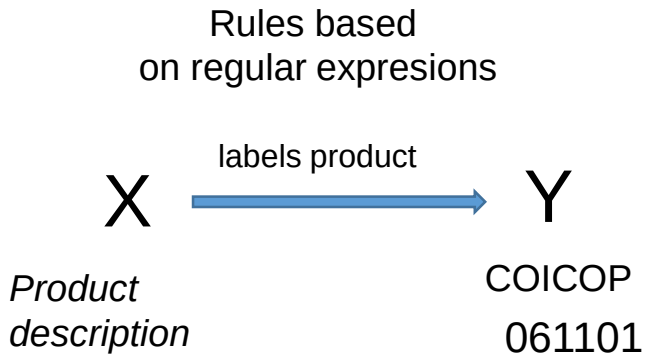
Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: Preprocessing, feature extraction, and

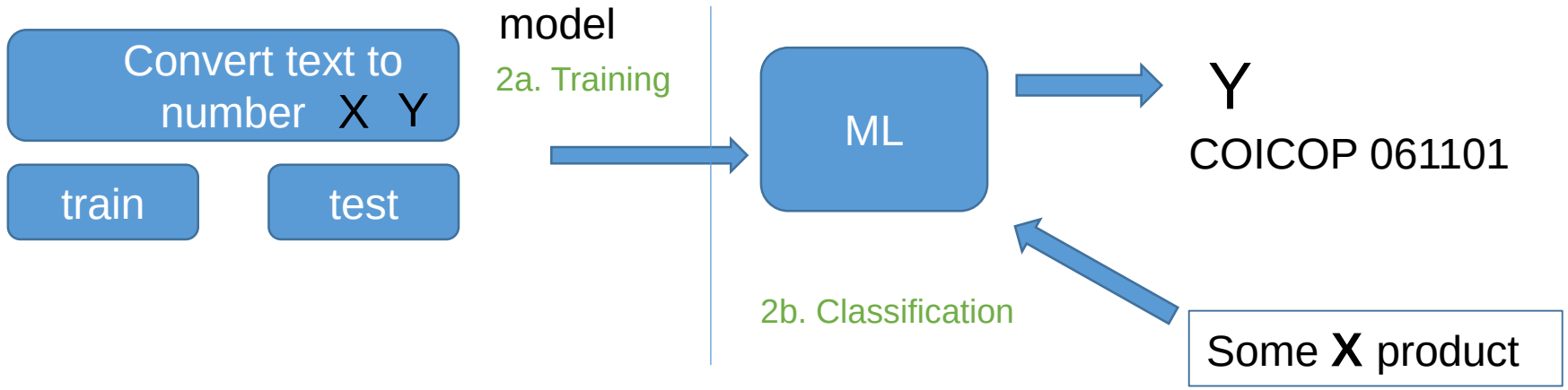


Types of text classification

1. Rule-based classification:
Use predefined rules that specify which text features correspond to certain categories.

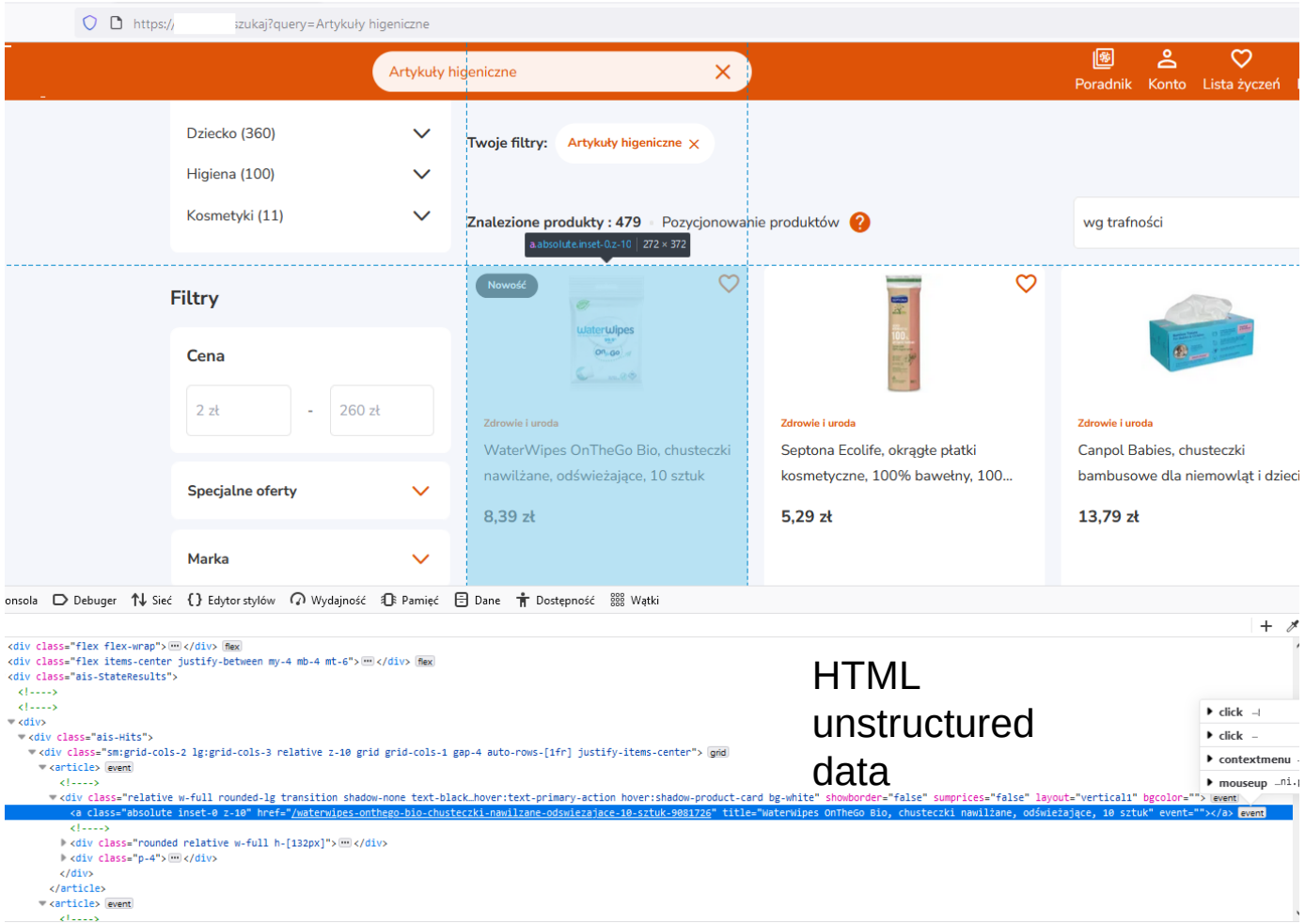


2. Machine learning (ML)
Models such as Naive Bayes, decision trees, SVM (Support Vector Machines), Logistic Regresion.



3. Hybrid
Combines both machine-based and rule-based approaches. It uses the rule-based system approach to create data tags and rules for machine learning models.

Data for classification



HTML unstructured data

Selenium

Raw data

BeautifulSoup

Structured data

Convert to numbers

ML classification

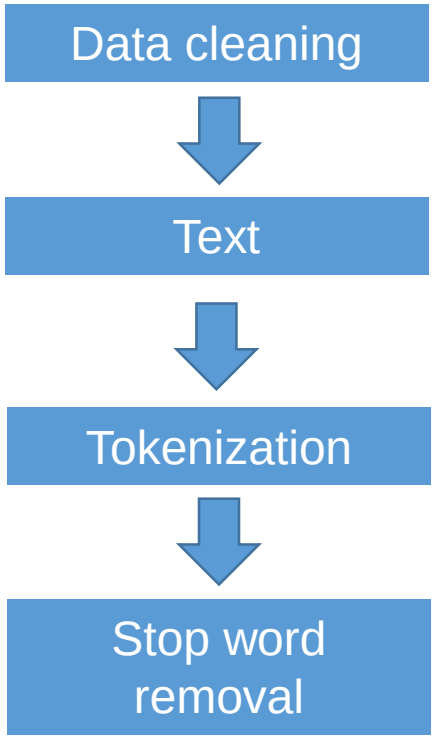
scikit learn

X → Y

Product name

COICOP 061101

Preparation of the text




```
def vectorization(self):
    vectorialize = CountVectorizer(
        token_pattern='\w\w+|[1-9]\.[1-9]\%|[1-9]\,[1-9]\%|[1-9]\.[1-9]|[1-9]\,[1-9]|[1-9]\%')
    vectorialize.fit(self.product['product_name'])

    self.vectorialize = vectorialize
    dat_set_vec = vectorialize.transform(self.product['product_name'])

    return dat_set_vec
```

Convert texts into numerical vectors

- One Hot Encoding - the vector contains 1 at the index corresponding to the word's position in the vocabulary, while all other elements are 0

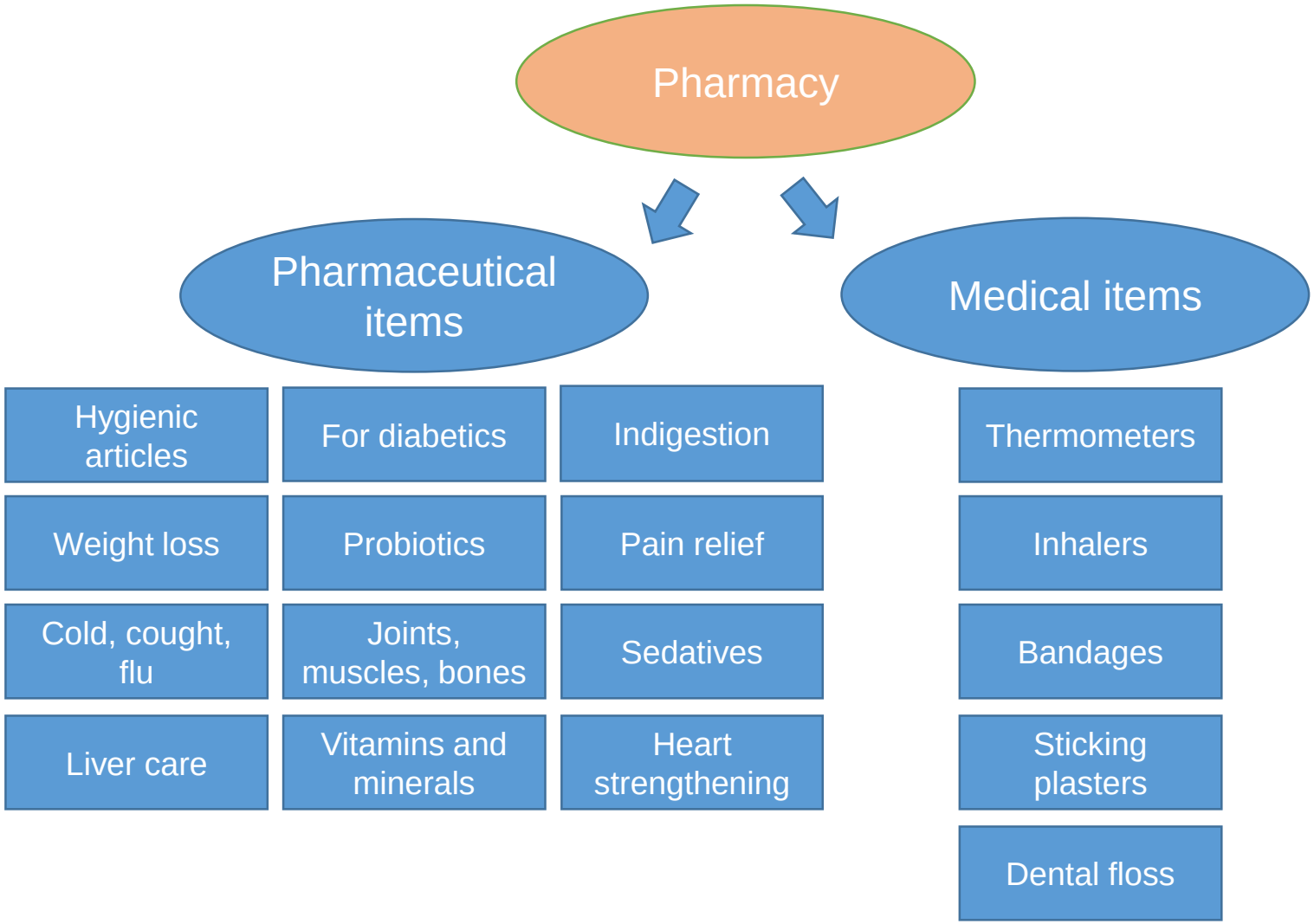
Document	Text	create	footprint	future	in	is	life	message	you	your
Document 1	You create your life	1	0	0	0	0	1	0	1	1
Document 2	Your life is your message	0	0	0	0	1	1	1	0	2
Document 3	You create footprint in life	1	1	0	1	0	1	0	1	0
Document 4	You create your future	1	0	1	0	0	0	0	1	1

Number of time the word "your" occurs in the Document 2

- BoW (Bag of Words) - every word is represented by a vector where each position corresponds to a unique word in the vocabulary

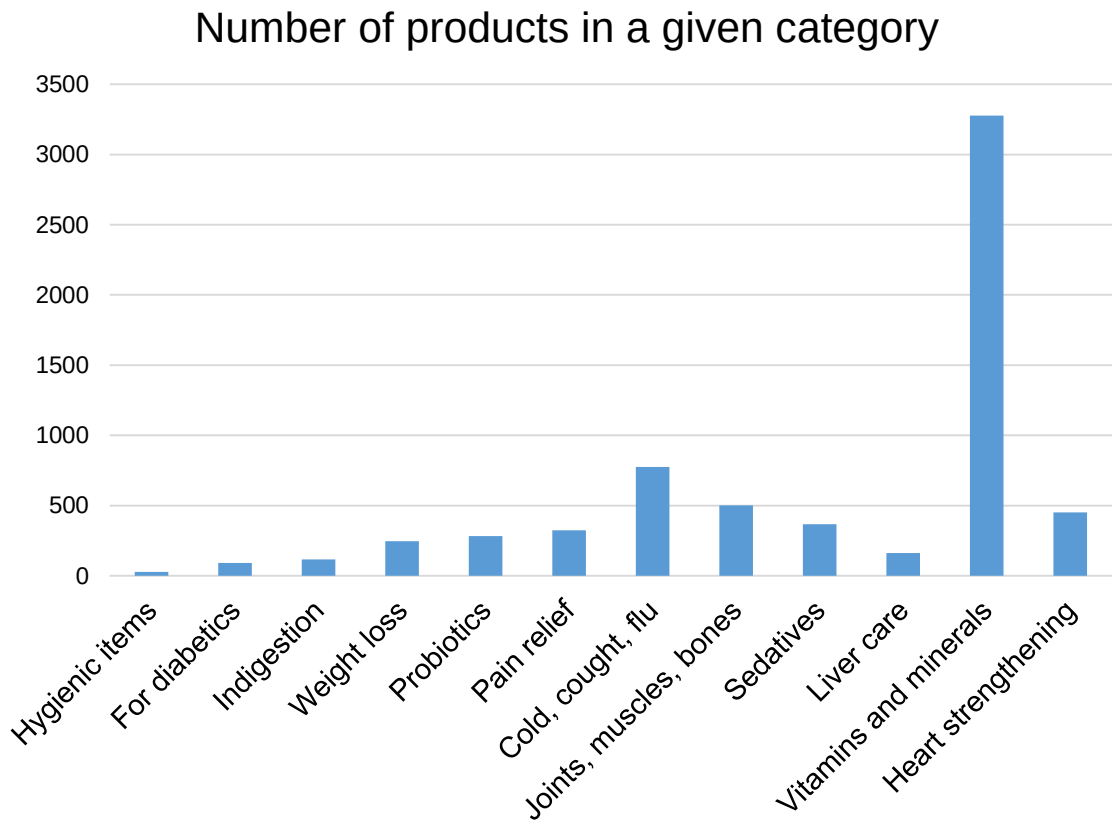
- TF-IDF (Term Frequency-Inverse Document Frequency)

Webscraping from online pharmacies



- 4 websites
- about 100-148 thousands rows per month
- data collection from 2019/09
- now 17 categories

Unbalanced classes



Category	Count
Hygienic items	27
For diabetics	92
Indigestion	117
Weight loss	246
Probiotics	282
Pain relief	323
Cold, cought, flu	775
Joints, muscles, bones	502
Sedatives	366
Liver care	161
Vitamins and minerals	3277
Heart strengthening	452
…	…

Web Application Pharmcy

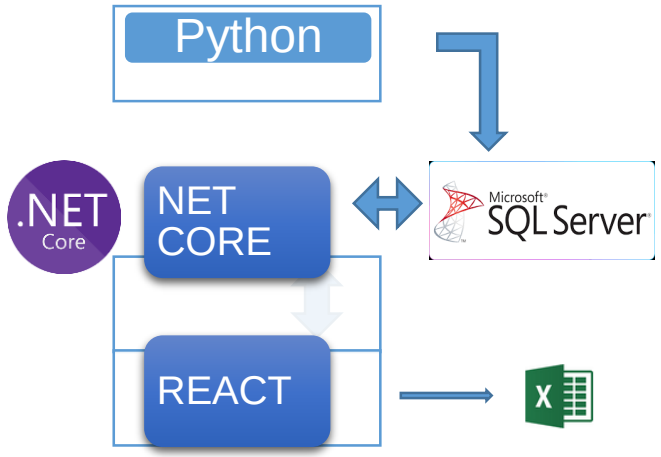
Wybrane produkty: 0

Export do EXCEL

Nazwa Towaru	Nazwa Kategorii	Coicop	Dokładność	Gramatu
Septolete Plus pastylki o smaku miodu i limonki	Przeziębienie i grypa	06110107	100	
Glukoza, proszek do przygotowania roztworu doustnego, 75 g	Inne	000000	100	

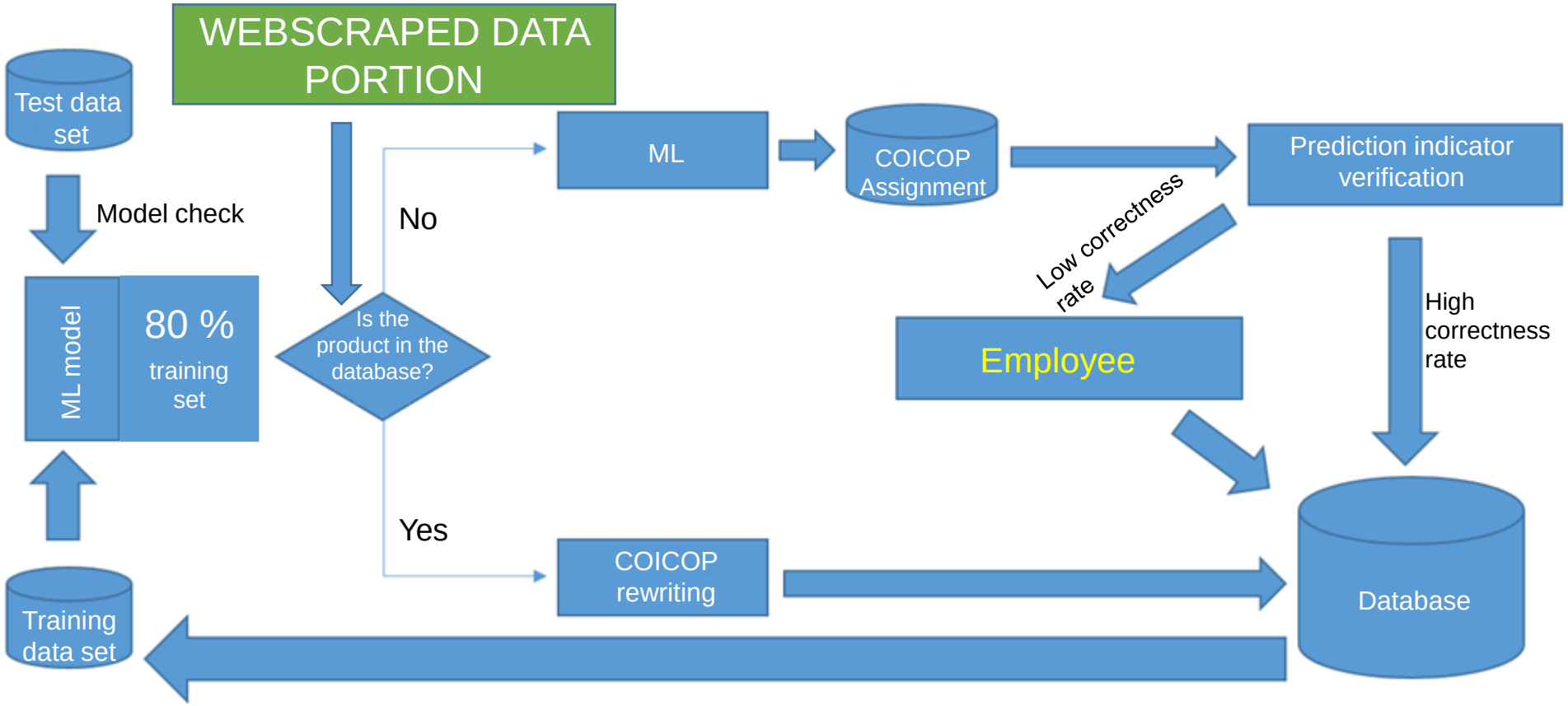
Backend (NET CORE) (engine of webstie aplication)
- read / edition of data in Microsoft SQL
- data processing and correction of COICOP classification

Fronted (REACT) (visible part of application)
- user interface



Technology stack

Processing data



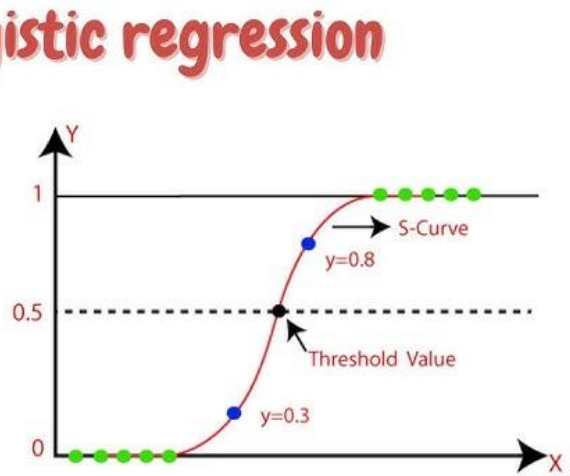
Classification: *COICOP*

ML models*: Logistic Regression, SVM, Naive Bayes*

Model training and evaluation

1. Data collection (webscraping)
2. Split data on train/test datasets on 80/20%
3. Convert text into numbers with CountVectorizer()
4. Model training
5. Model evaluation

COICOP classifiacation
(17 items)



LogisticRegression(C=13, solver='saga',multi_class='ovr', max_iter=300)

Results

Logistic Regression

	precision	recall	f1-score	support
000000	0.96	0.95	0.95	4637
06110101	0.90	0.79	0.84	243
06110102	0.85	0.73	0.78	321
06110103	0.79	0.65	0.71	271
06110104	0.78	0.63	0.70	632
06110105	0.96	0.86	0.91	608
06110106	0.93	0.94	0.93	482
06110107	0.93	0.90	0.91	1539
06110108	0.86	0.82	0.84	1277
06110109	0.83	0.73	0.78	485
06110110	0.93	0.73	0.82	437
06110111	0.81	0.93	0.87	4611
06110112	0.76	0.67	0.72	537
accuracy			0.88	16080
macro avg	0.87	0.79	0.83	16080
weighted avg	0.88	0.88	0.88	16080

Ranger

	precision	recall	f1-score	support
000000	0.91	0.96	0.93	4637
06110101	0.91	0.74	0.81	243
06110102	0.88	0.65	0.75	321
06110103	0.80	0.55	0.65	271
06110104	0.76	0.53	0.63	632
06110105	0.93	0.81	0.86	608
06110106	0.93	0.88	0.90	482
06110107	0.91	0.88	0.89	1539
06110108	0.87	0.77	0.81	1277
06110109	0.88	0.55	0.68	485
06110110	0.93	0.64	0.76	437
06110111	0.78	0.94	0.85	4611
06110112	0.85	0.56	0.67	537
accuracy			0.85	16080
macro avg	0.87	0.73	0.78	16080
weighted avg	0.86	0.85	0.85	16080

Statistics Poland

Thank you.