Statistics Poland

Big Data for Official Statistics

Jacek Maślankowski, Ph.D.
University of Gdańsk
Statistics Poland

AGENDA

- Introduction and objectives of the course
- Web scraping fundamentals
- Web scraping semi-structured data
- Machine learning fundamentals
- Machine learning and web data

Introduction and objectives of the course

Short overview of the agenda

Statistics Poland

Objectives

Show the fundamentals of the use of Big Data in official statistics using three different aspects of Big Data:

- web scraping,
- text mining and
- machine learning.

The course will be based on practical examples and exercises that will allow participants in better understanding the concept of the use of Big Data in official statistics.

Examples will be based on real issues of data gathering and processing for official statistics.

Detailed agenda (1/4)

- Web scraping – history, tools, types of web scraping
- Acquiring data from web – manual vs. automatic tools
- Quality issues in web scraping – sustainability, coverage and representativeness
- Examples and exercises

Detailed agenda (2/4)

- Web scraping semi-structured data
- Combining two different web data sources – de-duplication issues
- Examples and exercises

Detailed agenda (3/4)

- Machine learning fundamentals – supervised vs. unsupervised learning
- Examples of text and numeric data
- Text mining – processing high quality text data for machine learning
- Examples and exercises

Detailed agenda (4/4)

- Machine learning with web data – how to prepare a good training dataset
- Quality aspects of machine learning
- Examples and exercises

Statistics Poland

I. Web scraping fundamentals

- Web scraping – history, tools, types of web scraping
- Acquiring data from web – manual vs. automatic tools
- Quality issues in web scraping – sustainability, coverage and representativeness
- Examples and exercises

Self Assessment of ICT skills

Could you please estimate the level of your knowledge on Big Data and data analysis?

- ❑ (5) **Good** (I can use Python or R scripts, I understand regular expressions, machine learning and text mining, I lead Big Data projects)
- ❑ (4) **Rather good** (I have a theoretical knowledge on Big Data and participated in Big Data projects)
- ❑ (3) **Medium** (I know fundamentals of Big Data)
- ❑ (2) **Rather bad** (I don’t have any knowledge regarding Big Data)
- ❑ (1) **Bad** (I haven’t used a computer yet)

Terminology related to Big Data tools

Hadoop

Hive

Spark

Kafka

MapReduce

Python

NoSQL

Statistics Poland

Recommended reading

- **ESSNet Big Data**
  - https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/Main_Page
  - Methodology – IT Report, Quality Report, Methodology Report
    - https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/WP8_Reports,_milestones_and_deliverables
  - Literature Overview
    - https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/b/b5/WP8_Deliverable_8_1_LiteratureReview_20180109.pdf
- **UNECE Big Data Quality Framework**
  - A Suggested Framework for the Quality of Big Data, Deliverables of the UNECE Big Data Quality Task Team, December, 2014 https://statswiki.unece.org/display/bigdata/2014+Project?preview=%2F108102944%2F108298642%2FBig+Data+Quality+Framework+-+final-+Jan08-2015.pdf
- **General Papers**
  - Daas, P..H., Puts, M. J., Buelens, B., & Hurk, P. (2015). Big Data as a Source for Official Statistics, *Journal of Official Statistics*, *31*(2), 249-262. doi: https://doi.org/10.1515/jos-2015-0016

Big Data definition

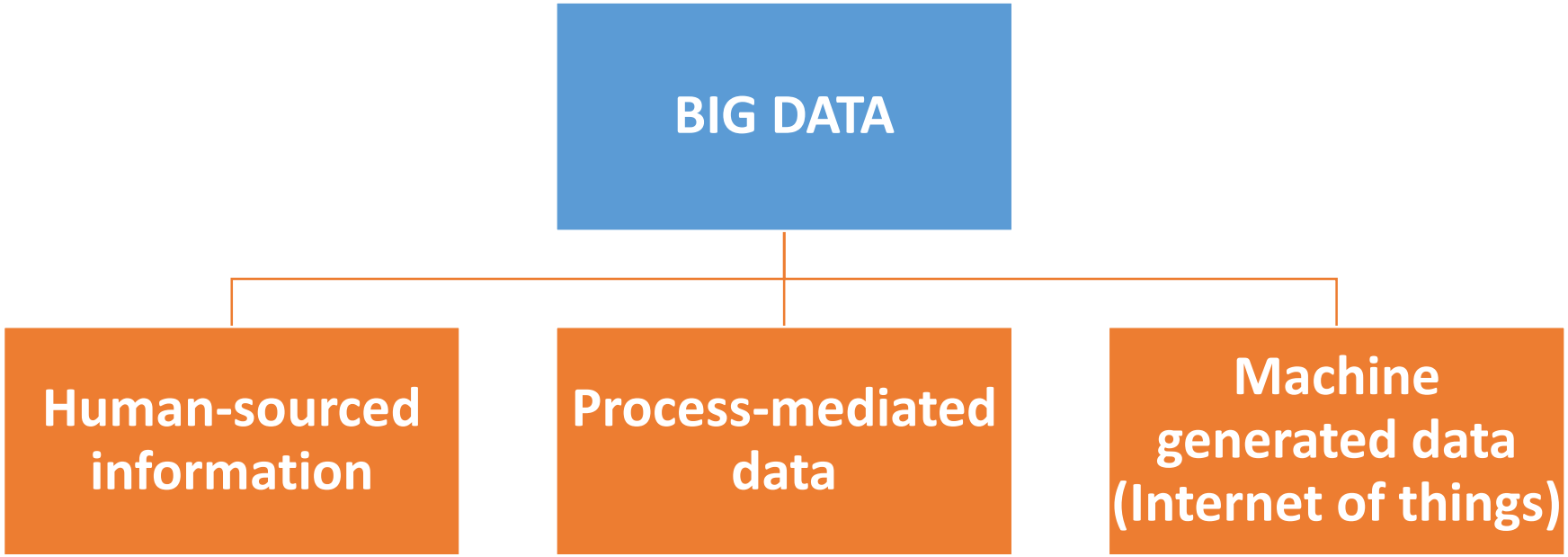
3V

- **V**olume (terabytes → petabytes → zettabytes)
- **V**ariety (structured → unstructured with structured)
- **V**elocity (batch data → streaming data)

+5V

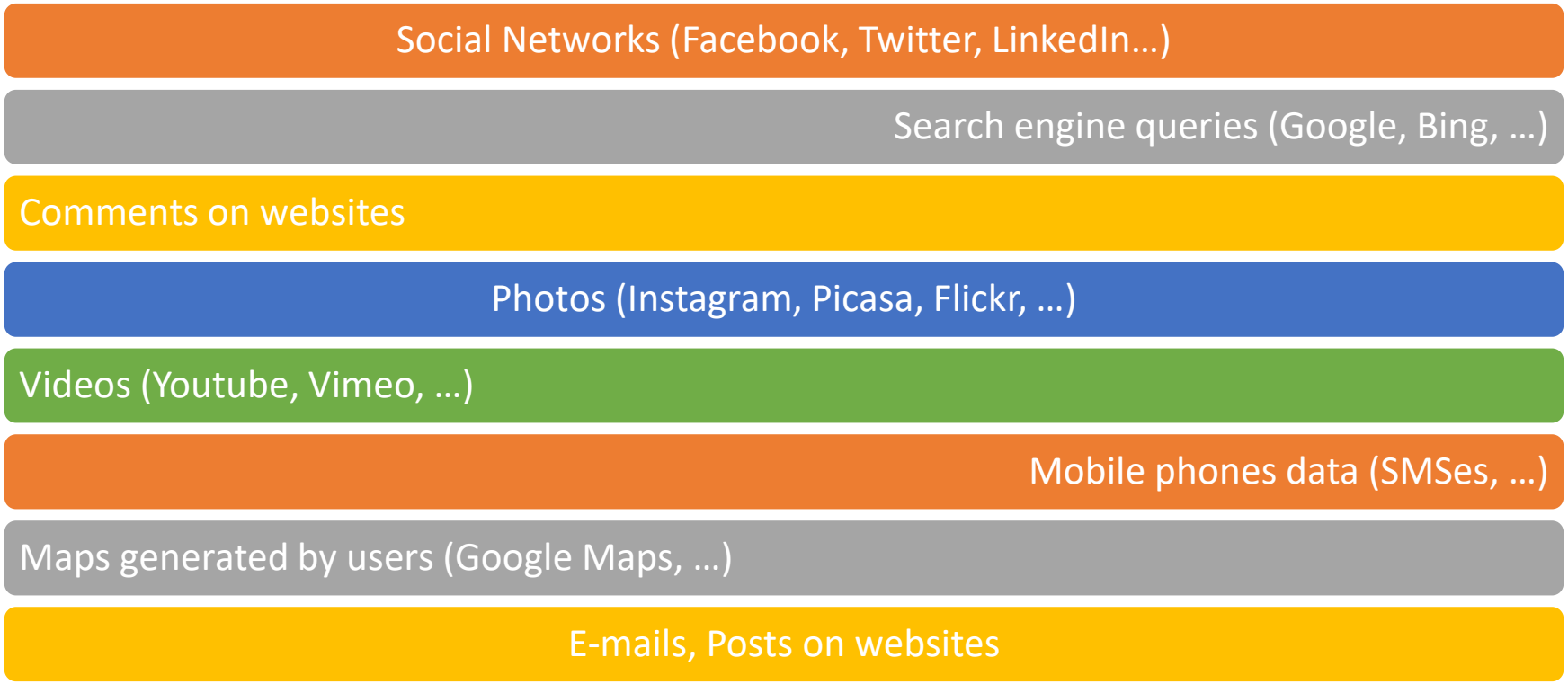
- Veracity (trust in data)
- Value (impact on decision process)

UNECE Big Data taxonomy



Statistics Poland

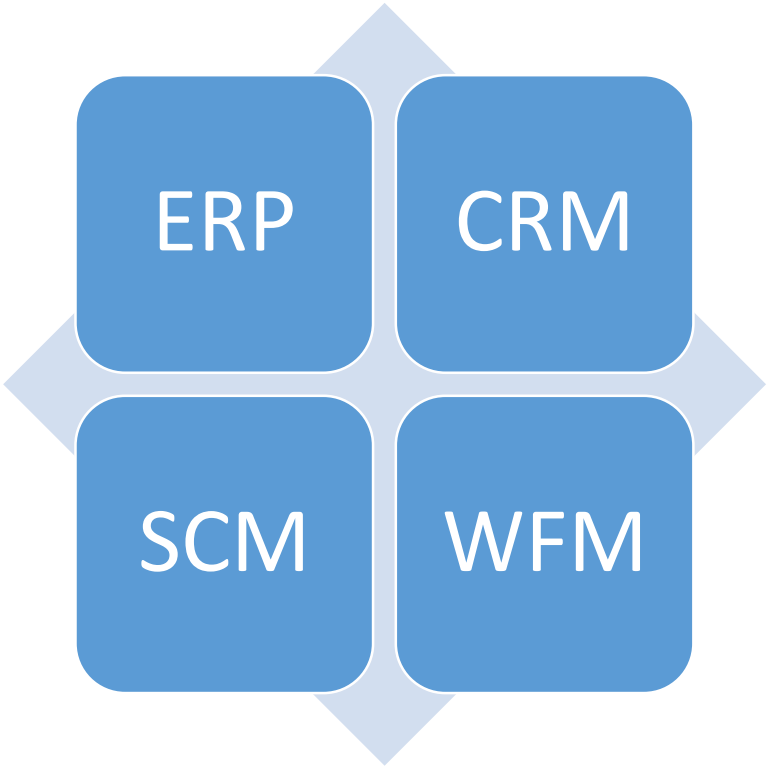
Human-sourced information



Statistics Poland

Process-mediated data (traditional business systems/transaction data)

- E-commerce (websites)
- Commercial transactions
- Banking records
- Stock prices
- Credit cards
- Medical records
- …



Statistics Poland

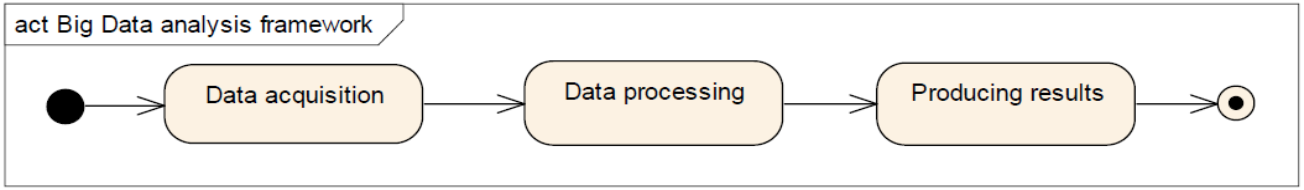
Machine generated data (Internet of things)

- Sensor data
- Weather/pollution sensors
- Traffic sensors/webcam
- Security videos/images
- …

- Tracking devices
- GPS systems
- Mobile phone location
- Satellite images
- Logs & Web logs
- …

Statistics Poland

Typical framework of Big Data



- ESSNet WP8 Methodology
  - Collect
  - Process
  - Analyse
  - Disseminate

- We have data or try to find data
- We don’t design the questionnaire
- Data driven vs. Goal driven vs. User driven approaches

Statistics Poland

Big Data and the quality

- **Input** – acquisition, or pre-acquisition analysis of the data
- **Throughput** – transformation, manipulation and analysis of the data
- **Output** – the reporting of quality with statistical outputs derived from big data sources

Source: UNECE Big Data Quality Framework

Example: Google Flu Trends

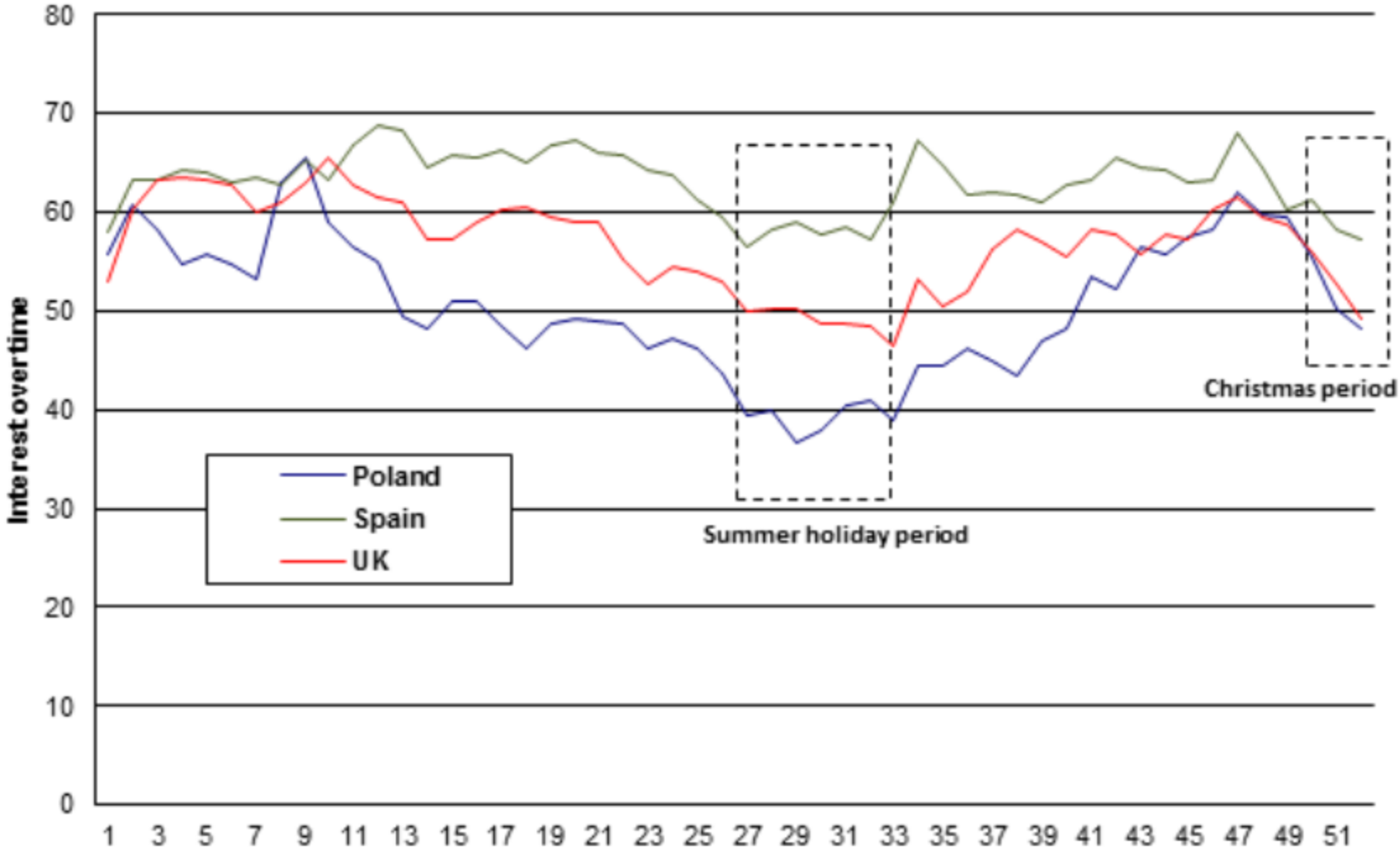
- Google Flu Trends
- Provides estimates of influenza activity for more than 25 countries
- https://www.google.org/flutrends/about/

Example: Google Trends

- https://www.google.com/trends/

- Discover and Correlate the trends on depression in different countries (ONS UK Use Case written in ESSNet Big Data WP7 paper)
- Try to analyse where the largest number of minority of people speaking Kashubian or Silesian language

Example: Google Trends – possible results



Source: Nigel Swier, WP7 document, Google Trends as a source for measuring sentiment and personal well-being

Statistics Poland

Example: Google Traffic

- How long does it take to go from one place to another?
- Distance in minutes between two streets/cities.
- Matrix of the distances in minutes:

	Point X	Point Y	Point Z
Point X	X	40 min.	55 min.
Point Y	30 min.	X	15 min.
Point Z	45 min.		X

What can be estimated with this data source?

Exercise: MCR – the way the data is stored

- HHHHHHHHHHAAABBWWWWWWWWWWWWWBBASSAAHHHHHHHHHHHHHH
- HHHHHHHHHHAAABBWWWWWWWWBCCCDEEDAAASSAAHHHHHHHHHHH
- HHHHHHHHHHAADDDEEESSEEECDDAAHHHHHHHHHHHHHHHHHHH

- H is home
- W is work
- A, B, C, D, E is a road between work and home
- S is shopping

Statistics Poland

Exercise: Find in Python

- # IMPORT LIBRARY
- **import pandas as pd**

- # LOAD THE CSV INTO DATAFRAME
- **df1=pd.read_csv("mcr.csv",delimiter=";")**

- # DISPLAY THE DATAFRAME
- **df1**

- # DATAFRAME DESCRIPTION
- **df1.describe()**

- # HOW MANY THERE IS ABC PATTERN # IN THE DATAFRAME
- **df1['mcrdata'].str.contains("W").value_counts()**

Statistics Poland

Exercise: Questions

1. What is the number of observations?
2. How many times was at home at any time of the day?
3. How many times was at work at any time of the day?
4. How many times traveled from home to work?
5. How many times traveled from work to home?
6. How many times traveled both from home to work as from work to home?
7. Is it a reliable data source for official statistics with one-hour data frequency?
8. What quality issues can be identified with this data source?

Quality issues

- **Coverage** – under- vs. over-coverage affect the representativeness and accuracy
- **Sustainability** of the data source
- **Comparability** over time
- **Linkability**

2. Web scraping semi-structured data

- Web scraping semi-structured data
- Combining two different web data sources – de-duplication issues
- Examples and exercises

Statistics Poland

What is web scraping?

ug.edu.pl

```
<!DOCTYPE html>
<html lang="pl" dir="ltr">
<head>
<meta name="format-detection" content="telephone=no" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=3, minimum-scale=1, user-scalable=yes" />
<link rel="shortcut icon" href="http://ug.edu.pl/sites/default/files/favicon.png" type="image/png" />
```

- Web scraping is a process of retrieving and processing information from websites.
- Web scrapers use similar methods of data processing as well known search engines and browsers.
- Web scrapers can extract information based on the structure of the website.

How the website is constructed?

- http://www.w3schools.com/tags

<a>	Hyperlink
<p>	Paragraph
<h?>	Header, where “?” 1-6
	Section
	Text line
/	Unorderer list/List item
<table>	Table
<tr> / <td>	Table row/table cell
<body>	Body – you see this in a browser
<html>	All HTML document

Statistics Poland

Example. Data inspection

- Data inspection on websites with web browsers.

Example. File robots.txt

- http://www.robotstxt.org/robotstxt.html

- Used to specify what can be done with the data on the website.

- Try to find the following files:
  - http://www.bok.or.kr/robots.txt
  - http://ug.edu.pl/robots.txt
  - http://www.stat.gov.pl/robots.txt
  - …

Best practice for official statistics – Netiquette by ONS UK and CBS NL (ESSNet Big Data WP2 example)

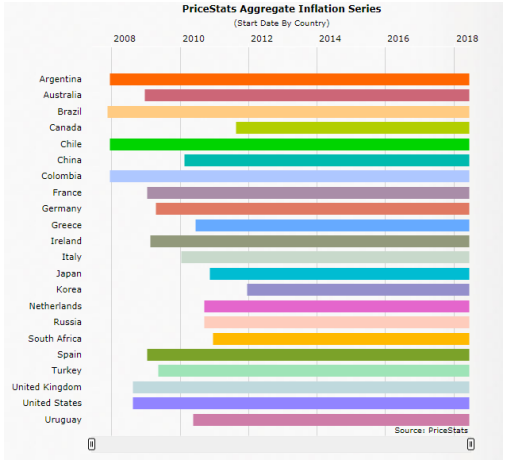
- Respect the '**robots.txt**' robots exclusion protocol and nofollow links
- **Identify** yourself in the user-agent string, and provide a means for the website to contact you, which could be via a link to a web-page
- Be **transparent** about your web-scraping activities, possibly by providing information on your website
- **Inform** website owners if a considerable amount of data is collected on a regular basis

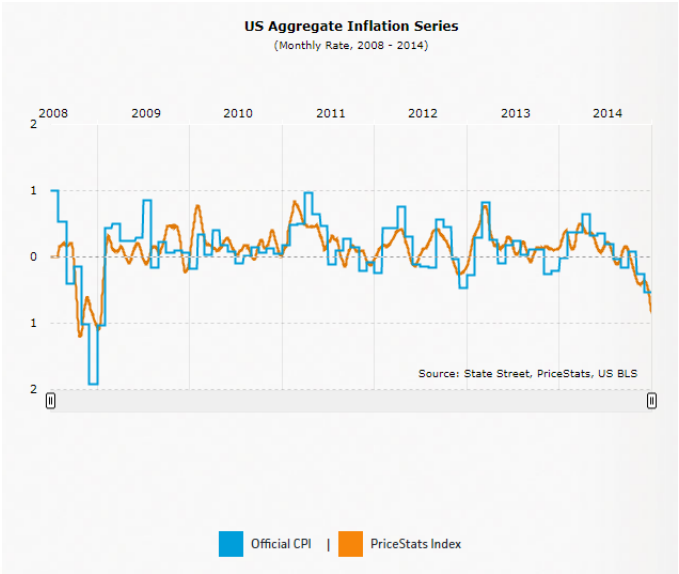
- Seek to minimise burden on website owners, for example:
  - By adding **idle time** between requests
  - Scrape at a time of day during which the website is **not under heavy load**
  - When crawling multiple domains, consider '**parallelising**' the crawl to avoid repeated requests to the same domain – submit a request to domain A, then domain B, then submit another request domain A
  - Do not crawl **sensitive areas** of a website URLs with /admin/ in, for example:
    - Only scrape data for the **production** of official statistics within the scope of your mandate, and do not re-use or distribute the data for any other purpose
    - Handle web-scraped data **securely** according to all relevant protocols and laws

- Source
  - https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/6/67/WP2_Deliverable_2_1_2017_02_15.pdf

Statistics Poland

Example. Web scraping prices





- Prices can be collected every day
- Result is comparable to CPI (Consumer Price Index)
- Data are more recent and available on request

- Source:
  - http://www.pricestats.com

Statistics Poland

Issues to consider

- Web scraping airline tickets prices
- Web scraping prices on several websites
- IP address and cookies of the scraper
- Sustainability of the data source

How it works?

1. Find a website with interesting data.
2. Analyse the structure of the website.
3. Write a script in Python, R or any other language.
4. Scrape the data.
5. Process them.

Be aware of the representativeness!

Tools used for webscraping

- www.urlitor.com/web-scraping
- Import.io
- Scrapy
- Imacros.net
- Apache Nutch
- Selenium
- …

- The list can be found here:
  - github.com/lorien/awesome-web-scraping
- Dedicated software can also be found on NSI's repositories.

Be aware of legal aspects

- There is Law on collecting the data from different databases
- The data should not be collected by robot and store if the owner does not allow
- Always check robots.txt

- Example of legal aspects (according to ESSNet Big Data)
  - https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/6/67/WP2_Deliverable_2_1_2017_02_15.pdf

Statistics Poland

Exercise.

- Find your favourite website (except social media) and check whether it is possible to web scrape this webpage regarding robots.txt content.

Exercise.

- What is used to present information at wzr.pl on employees?
  - Try to find containers/classes if possible.

- What is used to present information at ug.edu.pl on employees?
  - Try to find containers/classes if possible.

Exercise.

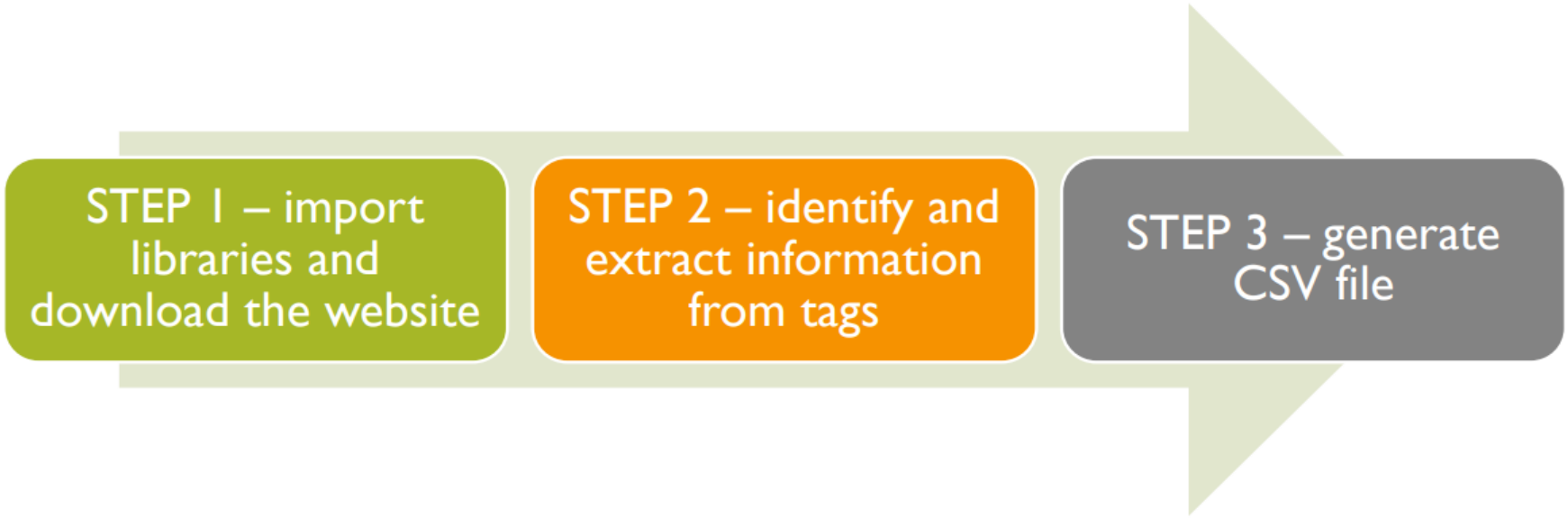
- Find a website with items/products and use page inspector to find elements responsible for the data.

Example. Python and web scraping

- Collect the data from different websites using requests.

```
1. import requests
2. page=requests.get("http://www.stat.gov.pl");
3. page.text
4. page.text[:10]
5. if "div" in page.text:
       print("DIV found")
```

Example. Web scraping



Step 1 – import libraries and download the website

1. import requests
2. from bs4 import BeautifulSoup
3. page=requests.get("http://wzr.pl/wydzial/index.php?str=121")

Step 2 – identify and extract information from DIVs

```
1. soup = BeautifulSoup(page.text, 'html.parser')
2. name_box = soup.findAll('div',
   attrs={'class':'pname plink'})
3. for i in range(len(name_box)):
4.   print (name_box[i].text.strip());
```

Exercise. Adding additional information.

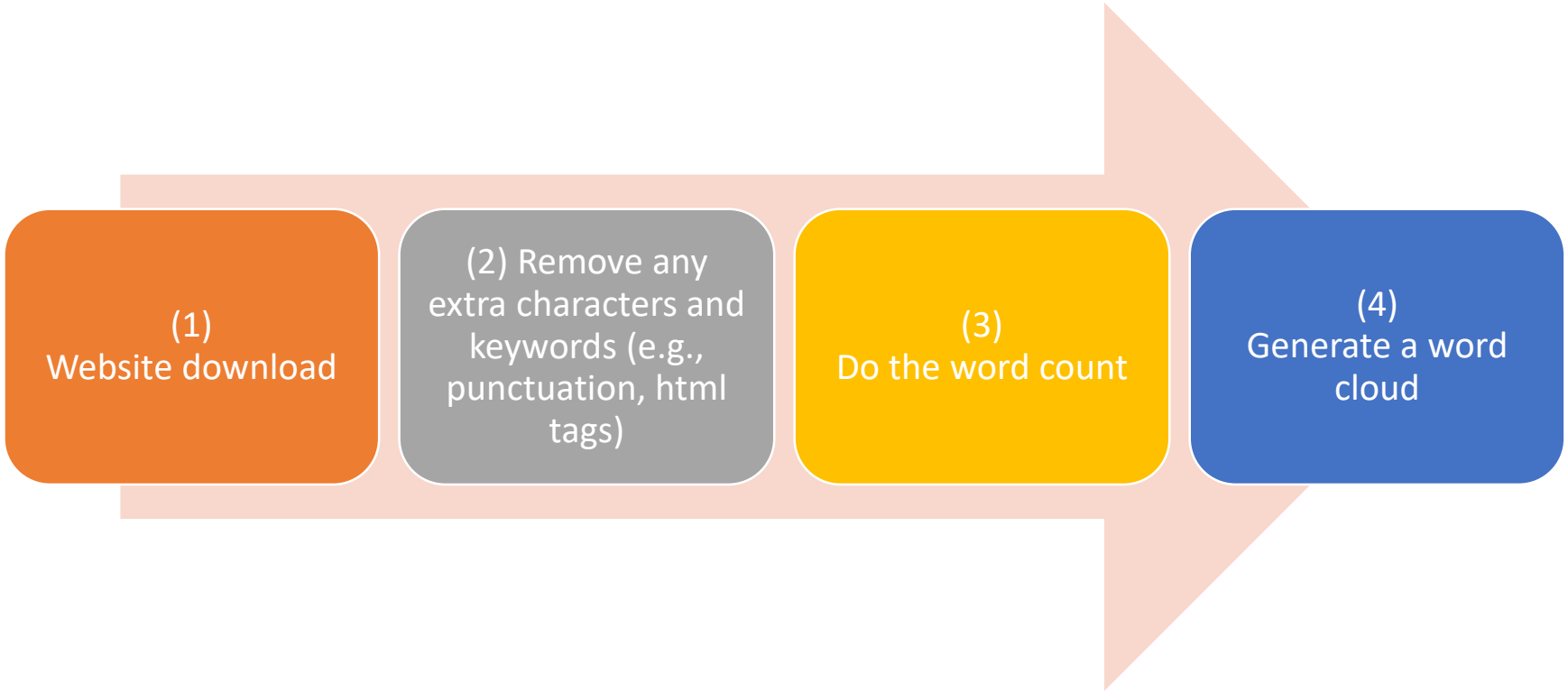
- Find DIV that store the data on job title.
- Modify the source code to store this information in job_title_box code.

Step 3. Write the output in CSV file

```
1. import csv
2. from datetime import datetime
3. with open('data.csv', 'a') as csv_file:
4.   writer = csv.writer(csv_file)
5.   for i in range(len(name_box)):
6.     writer.writerow
([name_box[i].text.strip(),job_title_box[i].text.strip(),datetime.now(
)])
```

Statistics Poland

Example. Word count analysis



Statistics Poland

STEP 1.

- import requests
- r = requests.get("http://yahoo.com")
- r.text[:100]

- Import a website and print the first 100 characters.

Statistics Poland

STEP 2.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(r.text, 'html.parser')
import re
def visible(element):
    if element.parent.name in ['style', 'script', '[document]', 'head', 'title']:
        return False
    elif re.match('<!--.*-->', str(element.encode('utf-8'))):
        return False
    return True
page_text_list = []
for t in filter(visible, soup.findAll(text=True)):
    page_text_list.append(t)
page_text = "".join(page_text_list)
page_text[:80]
page_words = page_text.split()
print(len(page_words), page_words[:10])
import string
page_words2 = [w.strip(string.punctuation).lower() for w in page_words if
len(w.strip(string.punctuation))>0]
print(len(page_words2), page_words2[:10])
```

- Remove any style, script, document, head and title content of the webpage
- Split the words into list and print them:
  - With punctuation
  - Without punctuation

Statistics Poland

STEP 3.

- from collections import Counter
- page_word_freq = Counter(page_words2).most_common()
- print(len(page_word_freq), page_word_freq[:10])

- Do the WORDCOUNT analysis and print the first 10 results

Statistics Poland

STEP 4.

- %pylab inline
- from os import path
- import matplotlib.pyplot as plt
- from wordcloud import WordCloud
- wordcloud = WordCloud(max_words=100).fit_words (page_word_freq)
- plt.imshow(wordcloud)
- plt.axis("off")

- Generate the WORDCLOUD with maximum 100 words

Statistics Poland

Exercise. Collecting and processing the data

- Change the webpage name in the previous example to any other website.
- What type of Big Data is it?
- Do the results satisfy you?

Stop words

- Stop words are are filtered out in processing of natural language data.
- E.g., the, and, or, an, …
- Go to https://www.ranks.nl/stopwords and find stopwords for your language.
- Are you happy with the results?

STEP 5. Stop words

- ENGLISH_STOP_WORDS = ["a", "about", "above", "across", … (more in example)
- word_freq_no_stop = [w for w in page_word_freq if w[0] not in ENGLISH_STOP_WORDS and not w[0].isdigit()]
- wordcloud = WordCloud(max_words=100).fit_words(word_freq_no_stop)
- plt.imshow(wordcloud)
- plt.axis("off")

- Exclude all stop words and generate the WORDCLOUD.

Example.

- Web scraping flight data

- What are the issues of web scraping such data?

Ident	Type	Destination	Departure	EstimatedArrival Time	Arrival
RYR1213	B738	Birmingham Int'l (BHX / EGBB)	So 11:05 CET	So 12:27 GMT	So 12:27 GMT
RYR2463	B738	London Stansted (STN / EGSS)	So 10:25 CET	So 11:27 GMT	So 11:27 GMT
RYR1641	B738	London Luton (LTN / EGGW)	Pt 21:45 CET	Pt 22:57 GMT	Pt 22:57 GMT

Summary – web scraping in official statistics

- Data on enterprises
  - NACE type of activity
  - URL
  - E-commerce, social media statistics or any other related to enterprise characteristics
- Prices
- Job vacancies
- Real Estate markets (price of properties)
- Tourism accomodation database
- …

Statistics Poland

Example.

- Find any web portal with job offers and try to identify job vacancies. Web scrap the data.
- Please check robots.txt before you scrap the data.
- Combine the dataset with another dataset.
- Check if you have duplicates in the data.

Example.

- Open any news portal and find comments below the news.
- Identify the tag and the class in which comment is stored.
- Two points of this step:
  - Download the website.
  - List all comments.
- Discuss the weaknesses of such approach.

Discussion on web scraping several data sources and quality issues

1. Coverage
2. Representativess
3. De-duplication
4. Accuracy

3. Machine learning fundamentals

- Machine learning fundamentals – supervised vs. unsupervised learning
- Examples of text and numeric data
- Text mining – processing high quality text data for machine learning
- Examples and exercises

What is machine learning?

- Supervised learning
- Unsupervised learning

+

- Semi-supervised Learning
- Reinforcement Learning

What is supervised learning?

- Let’s consider supervised training:
  - You teach the machine by preparing the training dataset
  - The machine is telling you the sentiment/type/anything you want

What can you teach?

- Text, e.g., sentiment analysis
- Pictures, e.g., crop types, people, gender
- Numbers, e.g., what type of flower it is based on dimensions, colour etc.

What is training dataset?

- I feel so bad today -> **sad**
- Tomorrow I start the vacation and feel so exhilirated -> **happy**
- I am so busy and tired -> **sad**
- We are going for holidays and I feel so lovely -> **happy**

Example of sentiment analysis

1. Happy
2. Sad
3. Scared
4. Depressed
5. Discouraged

What problems can you see in the sentiment analysis for the five categories like above?

Example. Using well known libraries

- Use MonkeyLearn to identify the sentiment of the sentence.
- https://app.monkeylearn.com
- Try to test several different sentences to see what is the result of them.

Aspects with text mining and sentiment analysis

1. Language issues
2. Extracting words and phrases from sentence
3. Tokenization and parsing
4. Lemmatization (better – good is a lemma)
5. Stemming (reduce to word stem, e.g., stem for stems, stemmed, stemming etc.)
6. Regular expressions
7. Analysis of expressions specific for the language
8. Entities
9. Stop words

Some aspects regarding text analysis and sentiment analysis

- Count number of specific words:
  - Good morning, my mood is not good today and I don’t feel good today.
  - It’s not bad today, really I am not unhappy.

Exercises on machine learning

Numeric data

Text data – sentiment analysis

Example. Machine learning with numeric data

(example from Python documentation, http://python.org, http://scikit-learn.org)

- Define the problem
- Prepare the data
- Assess the algorithm
- Improve the results
- Present the results

We will test the following algorithms…

- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN)
- Classification and Regression Trees (CART)
- Gaussian Naive Bayes (NB)
- Support Vector Machines (SVM)

Libraries

- import pandas
- from pandas.plotting import scatter_matrix
- import matplotlib.pyplot as plt
- from sklearn import model_selection
- from sklearn.metrics import classification_report
- from sklearn.metrics import confusion_matrix
- from sklearn.metrics import accuracy_score
- from sklearn.linear_model import LogisticRegression
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
- from sklearn.naive_bayes import GaussianNB
- from sklearn.svm import SVC

Load the data…

- url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
- names = ['sepal-length', 'sepal-width', 'petal-length',
  'petal-width', 'class']
- dataset = pandas.read_csv(url, names=names)

Dataset description

- print(dataset.shape)
- print(dataset.head(20))
- print(dataset.describe())
- print(dataset.groupby('class').size())

Data visualization

- dataset.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False)
- plt.show()

- dataset.hist()
- plt.show()

- scatter_matrix(dataset)
- plt.show()

Train the machine by splitting the dataset into proportion of 80:20

- # Split-out validation dataset
- array = dataset.values
- X = array[:,0:4]
- Y = array[:,4]
- validation_size = 0.20
- seed = 7
- X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=validation_size, random_state=seed)

Compare the result by using different algorithms

- # Test options and evaluation metric
- seed = 7
- scoring = 'accuracy’
- # Spot Check Algorithms
- models = []
- models.append(('LR', LogisticRegression()))
- models.append(('LDA', LinearDiscriminantAnalysis()))
- models.append(('KNN', KNeighborsClassifier()))
- models.append(('CART', DecisionTreeClassifier()))
- models.append(('NB', GaussianNB()))
- models.append(('SVM', SVC()))

- # evaluate each model in turn
- results = []
- names = []
- for name, model in models:
- kfold = model_selection.KFold(n_splits=10, random_state=seed)
- cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
- results.append(cv_results)
- names.append(name)
- msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
- print(msg)

Visualize the data with algorithm comparison

- # Compare Algorithms
- fig = plt.figure()
- fig.suptitle('Algorithm Comparison')
- ax = fig.add_subplot(111)
- plt.boxplot(results)
- ax.set_xticklabels(names)
- plt.show()

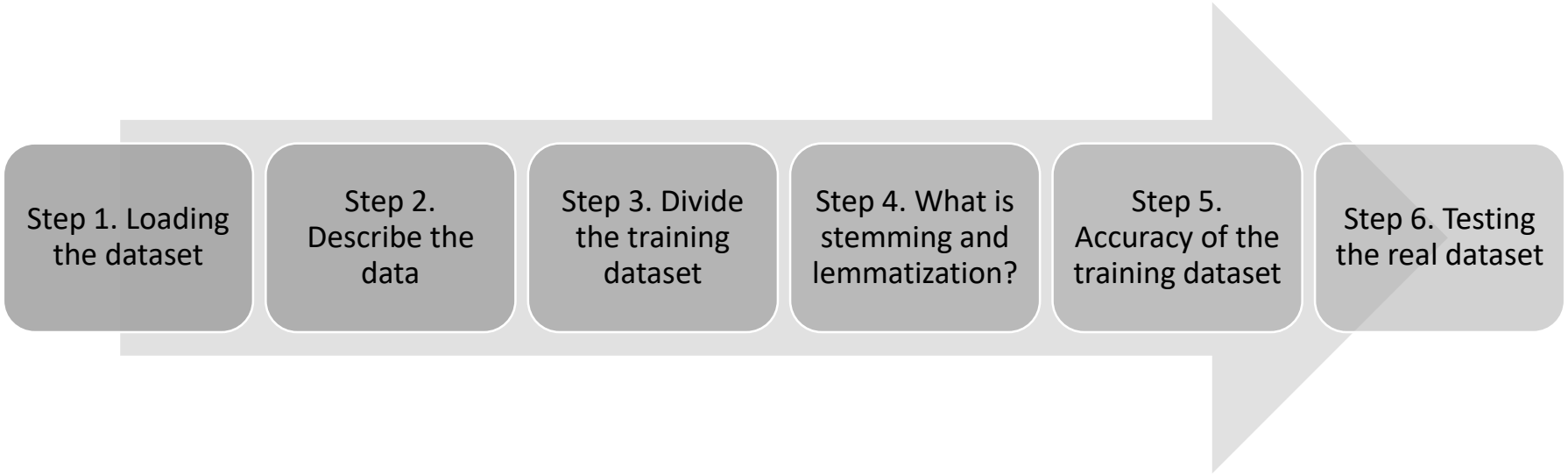
- # Make predictions on validation dataset
- knn = KNeighborsClassifier()
- knn.fit(X_train, Y_train)
- predictions = knn.predict(X_validation)
- print(accuracy_score(Y_validation, predictions))
- print(confusion_matrix(Y_validation, predictions))
- print(classification_report(Y_validation, predictions))

Exercise. Change the proportion of the training:testing dataset.

- ***validation_size = 0.20***

- Does it improve the results or not and why?

Example. Sentiment analysis
Six simple steps with machine learning on text data



Step 1. Loading the dataset

- ############## STEP 1
- ############## LOADING THE DATASET
- import pandas as pd
- import numpy as np
- import re, nltk
- from sklearn.feature_extraction.text import CountVectorizer
- from nltk.stem.porter import PorterStemmer
- from sklearn.cross_validation import train_test_split
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics import classification_report
- from collections import Counter
- test_data_file_name='testing_data_JM.txt'
- train_data_file_name='training_data_JM.txt'
- test_data_df = pd.read_csv(test_data_file_name, header=None, delimiter=";")
- test_data_df.columns = ["Text"]
- train_data_df = pd.read_csv(train_data_file_name, header=None, delimiter=";")
- train_data_df.columns = ["Id","TypeText","Text","Type"]
- TypeText=['positive','negative','indeterminate']

Step 2. Describe the data

- ############## STEP 2
- ############## dataset description

- ############## STEP 2a
- test_data_df.head()

- ############## STEP 2b
- train_data_df.head()

- ############## STEP 2c
- test_data_df.shape

- ############## STEP 2d
- train_data_df.shape

Statistics Poland

Step 3. Divide the training dataset

- ############## STEP 3
- np.mean([len(s.split(" ")) for s in train_data_df.Text])
- stemmer = PorterStemmer()
- def stem_tokens(tokens, stemmer):
- ....
- return stemmed
- def tokenize(text):
- .....
- return stems

```
vectorizer = CountVectorizer(
    analyzer = 'word',
    tokenizer = tokenize,
    lowercase = True,
    stop_words = 'english',
    max_features = 85
)
)
y_pred = log_model.predict(X_test)
```

- corpus_data_features = vectorizer.fit_transform(
- train_data_df.Text.tolist() + test_data_df.Text.tolist())
- corpus_data_features_nd = corpus_data_features.toarray()
- corpus_data_features_nd.shape
- vocab = vectorizer.get_feature_names()
- dist = np.sum(corpus_data_features_nd, axis=0)
- X_train, X_test, y_train, y_test = train_test_split(
- corpus_data_features_nd[0:len(train_data_df)],
- train_data_df.Type,
- **train_size=0.85,**
- random_state=1234)
- log_model = LogisticRegression()
- log_model = log_model.fit(X=X_train, y=y_train

Statistics Poland

Step 4. What is stemming and lemmatization?

- ############## STEP 4
- ############## WHAT IS STEMMING AND LEMATIZATION?
- print(vocab)

Step 5. Accuracy of the training dataset

- ################ STEP 5
- ################ TESTING TRAINING DATASET
- print("Testing the training dataset accuracy...")
- print(classification_report(y_test, y_pred))

Step 6. Testing the real dataset

- ############## STEP 6
- ############## TESTING THE REAL DATASET
- log_model = LogisticRegression()
- log_model = log_model.fit(X=corpus_data_features_nd[0:len(train_data_df)], y=train_data_df.Type)
- test_pred = log_model.predict(corpus_data_features_nd[len(train_data_df):])

- import random
- spl = random.sample(range(len(test_pred)), len(test_pred))
- purpose=[]
- for text, type in zip(test_data_df.Text[spl], test_pred[spl]):
- print (TypeText[type-1],':', text)
- purpose.append(TypeText[type-1])

- print("The following life satisfaction sentiments were identified:\n")
- c = Counter(purpose)
- for letter in c:
- print ('%s: %d' % (letter, c[letter]))

Statistics Poland

Exercise.

- Prepare a training dataset for the following data:
  - Happy
  - Sad
  - Not determined
- Test the training dataset.

Exercise.

- Try to web scrap the web page and give any possibility of the use of machine learning.

Example.

Consider social media as a data source.

Is it possible to precisely determine the sentiment of Twitter tweet or Facebook post?

Discussion

What can be achieved by the use of machine learning with numeric and text data?

Are we aware of problems with the preparation of the training dataset?

Are we ready to disseminate the data as the official statistics?

What is the risk in terms of data quality with machine learning?

4. Machine learning and web data

- Machine learning with web data – how to prepare a good training dataset
- Quality aspects of machine learning
- Examples and exercises

Social media presence – use case PL: general overview
(available at: https://github.com/jmaslankowski for ESSNet Big Data WP2)

(1) Web scraping
- HTML File
- Extracting all links to find social media links
- Next iteration if the link is not present

(2) Twitter API
- Scrap the tweet
- Process the post/tweet by Machine Learning algorithm

(3) Machine Learning
- Classify the tweet:
  - based on C11 (ICT 2015)

(1)
```
with open ('wp2_social.csv','a') as plikcsv:
    kolumny=['URL','Facebook','Twitter','Youtube','LinkedIn','Instagram','GooglePlus']
    zapis=csv.DictWriter(plikcsv,delimiter=';',dialect=csv.excel,fieldnames=kolumny)
```

(2)
```
trening.target_name=
 ['others',
  'recruitment',
  'marketing',
  'enterprise image',
  'commercials']
```

(3)

Use of Social Media

Enterprises using social media are considered those that have a user profile, an account or a user licence depending on the requirements and the type of the social media.

C10. Does your enterprise use any of the following social media? (not solely used for paid adverts) *(add national examples; replace existing examples if necessary)*	Yes	No
a) Social networks (e.g. Facebook, LinkedIn, Xing, Viadeo, Yammer, etc.)	☐	☐
b) Enterprise's blog or microblogs (e.g. Twitter, Present.ly, etc.)	☐	☐
c) Multimedia content sharing websites (e.g. YouTube, Flickr, Picasa, SlideShare, etc.)	☐	☐
d) Wiki based knowledge sharing tools	☐	☐

The following question (C11) should only be answered if any of the above social media is used (i.e. C10 has at least one "Yes").

C11. Does your enterprise use any of the above mentioned social media to:	Yes	No
a) Develop the enterprise's image or market products (e.g. advertising or launching products, etc)	☐	☐
b) Obtain or respond to customer opinions, reviews, questions	☐	☐
c) Involve customers in development or innovation of goods or services	☐	☐
d) Collaborate with business partners (e.g. suppliers, etc.) or other organisations (e.g. public authorities, non governmental organisations, etc.)	☐	☐
e) Recruit employees	☐	☐
f) Exchange views, opinions or knowledge within the enterprise	☐	☐

https://circabc.europa.eu/sd/a/7956316e-50f6-4f14-a144-055cb8af4901/Questionnaire%20ENT2015.pdf

C10. Does the Website have any of the following?	Yes	No
a) Description of goods or services, price lists	☐	☐
*[8] b) Online ordering or reservation or booking, e.g. shopping cart	☐	☐
c) Possibility for visitors to customise or design online goods or services	☐	☐
d) Tracking or status of orders placed	☐	☐
e) Personalised content in the website for regular/recurrent visitors	☐	☐
f) Links or references to the enterprise's social media profiles	☐	☐
g) Advertisement of open job positions or online job application - Optional	☐	☐

Does your enterprise use any of the following social media? (not solely used for paid adverts) *(additional examples; replace existing examples if necessary)*	Yes	No
a) Social networks (e.g. Facebook, LinkedIn, Xing, Viadeo, Yammer, etc.)	☐	☐
b) Enterprise's blog or microblogs (e.g. Twitter, Present.ly, etc.)	☐	☐
c) Multimedia content sharing websites (e.g. YouTube, Flickr, Picasa, SlideShare, etc.)	☐	☐
d) Wiki based knowledge sharing tools	☐	☐

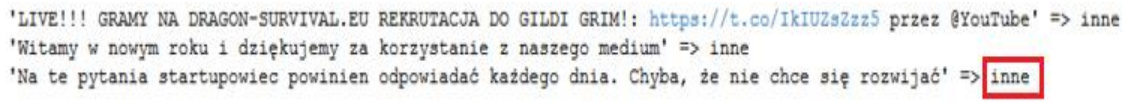
https://circabc.europa.eu/sd/a/a39ae859-8a16-4306-8020-ae06d3df3c91/Questionnaire%20ENT%202016.pdf

Statistics Poland

Results and result sets

- Social media URL finding

- Machine learning



Statistics Poland

Algorithms
and technological obstacles

- Difficulties with accessing the data, e.g., from Facebook
- Some social media links does not exists, e.g., Present.ly
- The classification should reflect the real usage of social media
- Providing the reliable training set
- Improving Machine Learning algorithms
  - Multinomial Naive Bayes – MultinomialNB()

```
Out[131]: 0.7142857142857143
```

  - Stochastic Gradient Descent – SGDClassifier()

```
Out[132]: 0.8095238095238095
```

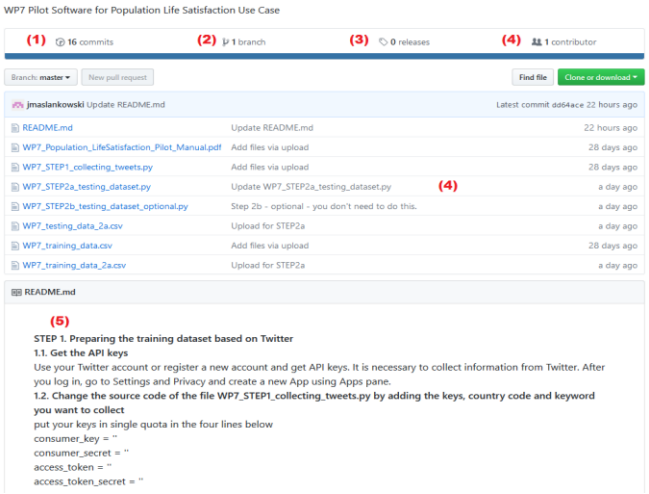
  - …

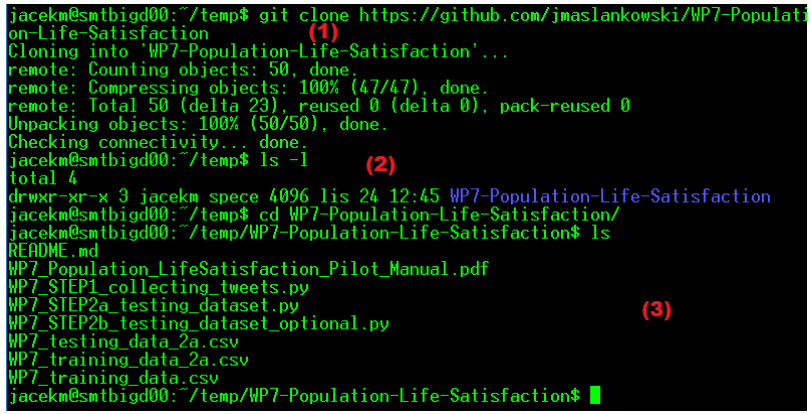
	precision	recall	f1-score	support
	0.88	0.88	0.88	16
	0.50	0.50	0.50	4
	1.00	1.00	1.00	1
	0.81	0.81	0.81	21

Github Repositories

(available in: https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/1/10/WP8_Deliverable_8.3_IT_Report_2018_03_05_final.pdf)

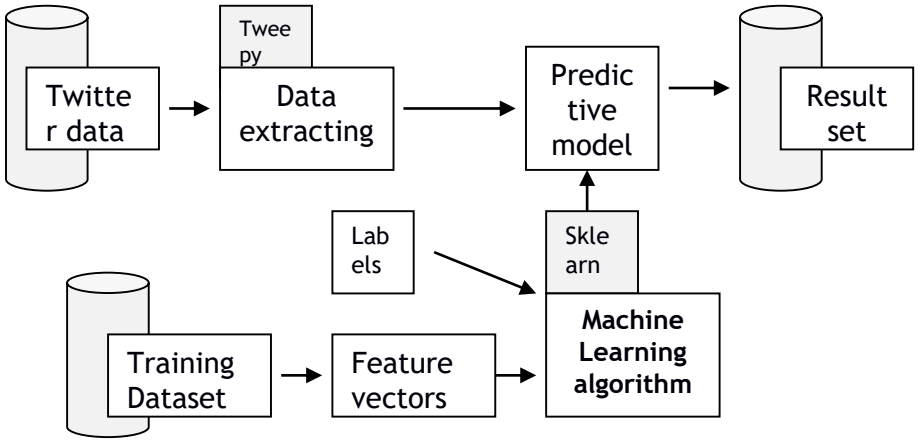
No.	Name	Link	Main features
1	Awesome Official Statistics software	https://github.com/SNStatComp/awesome-official-statistics-software	The list of useful statistical software with links to other GitHub repositories, by CBS NL
2	ONS (Office for National Statistics) UK Big Data team	https://github.com/ONSBigData	Various software developed by ONS UK Big Data Team
3	ONS (Office for National Statistics) UK Data Science Campus	https://github.com/datasciencecampus	Various software developed by ONS Data Science Campus Team
4	ESTP Big Data course software	https://github.com/SNStatComp/ESTPBD	Various software developed for the ESTP Big Data training courses





Framework that is used in PL Population Use Case to scrap and process Twitter data

(available at: https://github.com/jmaslankowski for ESSNet Big Data WP7)

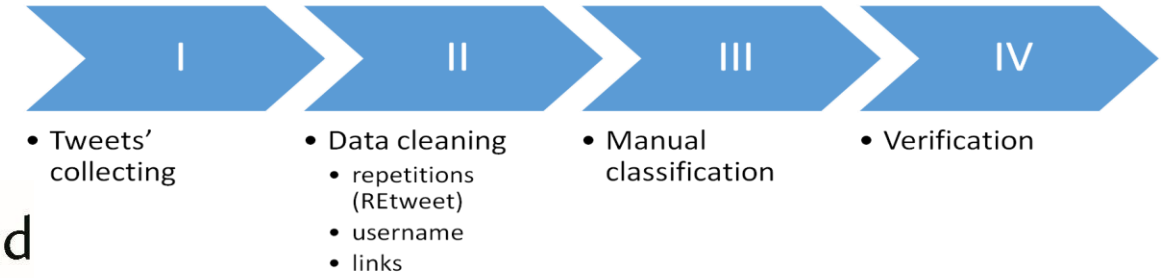


We use the following classification based on emotional states from European Social Survey, Social Cohesion Survey and EU Statistics on Income and Living Conditions (EU-SILC):

- szczęśliwy (happy),
- neutralny (neutral),
- spokojny (calm),
- zdenerwowany (upset),
- przygnębiony (depressed),
- zniechęcony (discouraged),
- nie wiem (indeterminate).

The instruction on how to prepare a good training dataset was the next step of the meeting.

We need to divide this task into four stages, presented in the picture below.



Statistics Poland

AGRICULTURE – Estimation of Agricultural statistics – pilot case study on crop types based on satellite data (based on ESSNet Big Data WP7)

Responsibility: PL – coordinator, supported by IE.

Data sources: Satellite images, administrative data, in situ surveys.

Methodology:

combining data – data fusion on radar and optical remote sensing data;

data comparison with traditional surveys e.g. FSS;

combining data – administrative data sources with satellite data.

The goal of the case study: Crop type: look at the types of crops being grown and see if we can tell this accurately from the imagery; analysis of possibilities of using satellite images.

Plan of Combining Datasets: Data fusion – combining data sources by spatial reference.

Main benefits and value added for official statistics:

Increase the quality of the agricultural surveys;

Decrease of respondents burden;

More detailed data published by official statistics;

Potential decrease of the cost of conducting surveys.

Statistics Poland

Experimental statistics for Official Statistics

- http://ec.europa.eu/eurostat/web/experimental-statistics/
- https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/Experimental_big_data_statistics
- https://www.cbs.nl/en-gb/our-services/innovation

Discussion

1. What is the usefulness of presented methods?
2. What quality dimensions we can assess?
3. Are there any possibilities in applying such methods to official statistics?

Summary: additional data sources

AIS data

Mobile phone data

Smart meters data

…

What is the obstacle to access such a dataset?

What is the value added of the energy or water consumption received hour by hour?

Statistics Poland

Summary: exercise integrating web scraping and machine learning methods

Web scrape comments from any news portal.

Try to make analysis of them regarding sentiment analysis.

What can be observed in such a dataset?

What are the quality issues?

Is it possible to combine this dataset with another?

Discussion

Q&A

Statistics Poland

Statistics Poland

Thank you!

j.maslankowski@stat.gov.pl