Mariusz Gil

Machine Learning for a

RESCUE

mariuszgil

Source Ministry

CLIENT PROBLEM

1M BACKLINKS CLASSIFY THEM

OK

NOT OK

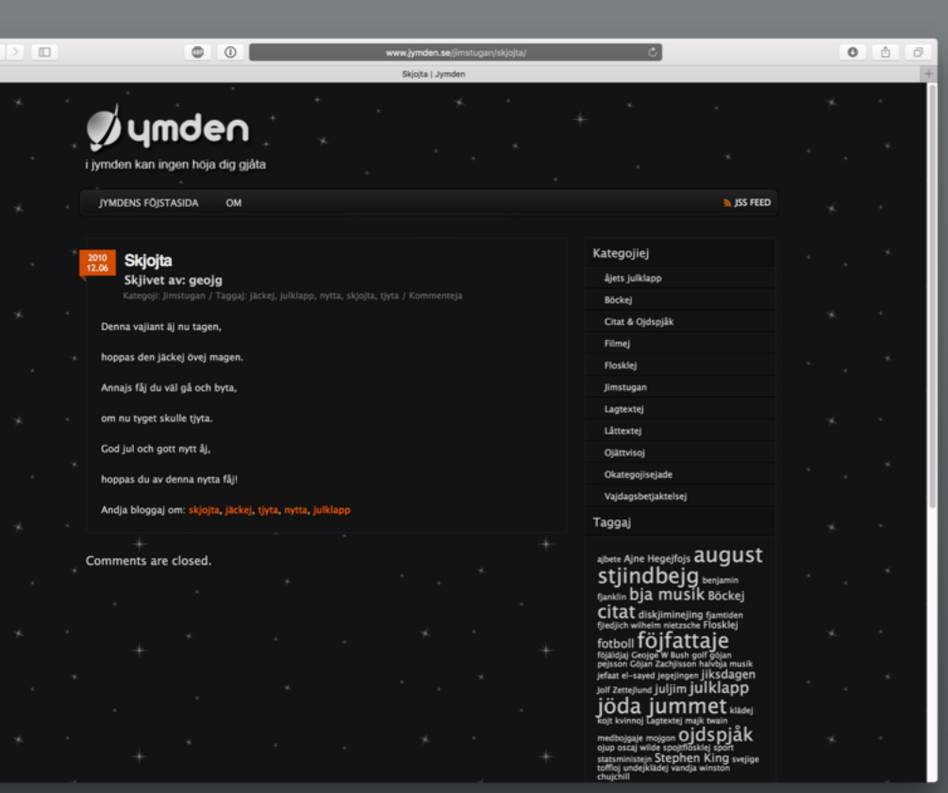
I DON’T CARE

OK

NOT OK

I DON'T CARE

OK



NOT OK

I DON’T CARE

naTemat.pl
A to się Duda zdziwi! Według kancelarii prezydenta, ułaskawienie bez prawomocnego wyroku jest...

naTemat.pl
Farsa przy wyborze członków kolegium IPN. Prokurator z PRL przesłuchiwał... legendy opozycji

INNPoland.pl
Co z tą Brytanią? My błądzimy we mgle, Internet już wie

MamaDu.pl
Pokazała ciążowy brzuch nie była. Blogerka zdrad internetu

Grupa naTemat

Jak Google chciało zamknąć biznes Polakom. Kolejny powód, żeby ograniczyć wszechwładzę wyszukiwarki

TOMASZ MOLGA

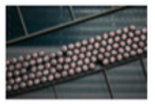
dawno temu

Michał Jaskólski współtwórca porównywarki Nokaut.pl, która już dwa lata temu doświadczyła potęgi internetowej korporacji Google • YouTube.

7,7 mln złotych strat, zwolnienie 30 ze 120 zatrudnionych pracowników i internetowy serwis, na który po interwencji wielkiego brata Google nie zajrzał pies z kulawą nogą. Tak wszechwładny Google może pogrywać ze swoimi przeciwnikami. Tyle tylko, że ta historia wydarzyła się naprawdę i

ZOBACZ TAKŻE:

Czy bank może odmówić przeniesienia usługi? Tutaj odpowiedzi


Znaleźliśmy idealny zawód!

Mężczyzna niejedno ma imię. Sprawdź, jakim typem faceta jesteś

Dlaczego nie chcemy zmieniać usług finansowych? Sprawdzamy

10 najlepszych komedii, jakie musisz obejrzeć z kolegami!

5 stereotypowych przekonań, którymi tłumaczysz brak oszczędności

T(URL) → [1, 2, 3, ...]

1ST APPROACH

IF-OLOGY

UGLY CODE FOR POC

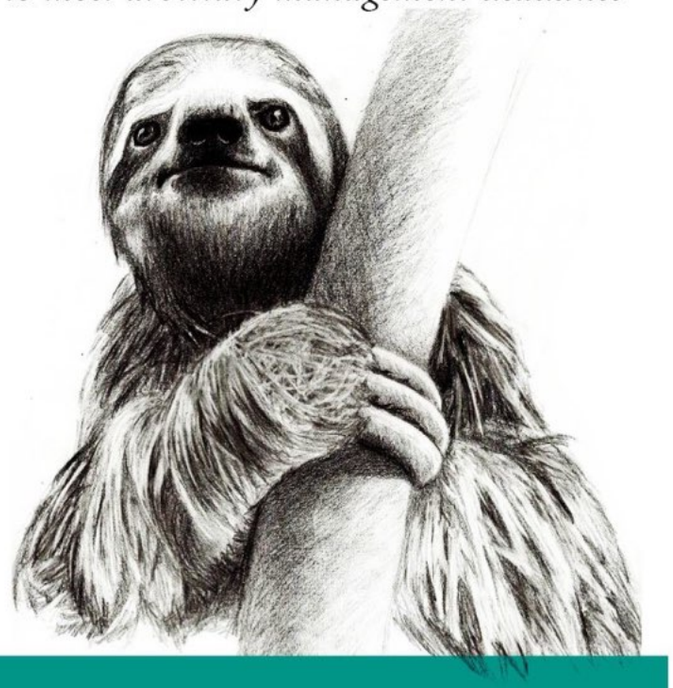
I DON' KNOW

2ND APPROACH

NAIVE MACHINE LEARNING

DATA
SEND TO
ML
TASK
CALCULATE
RESULTS

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

Software can be chaotic, but we make it work

Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev

DOING WITHOUT KNOWING IS A. . .

RECIPE FOR A FAILURE

3RD APPROACH, FINAL

DATA ORIENTED MACHINE LEARNING WORKFLOW

A COMPUTER PROGRAM
IS SAID TO LEARN FROM **EXPERIENCE E**
WITH RESPECT TO SOME CLASS OF **TASKS T**
AND **PERFORMANCE MEASURE P**
IF ITS PERFORMANCE AT TASKS IN T,
AS MEASURED BY P,
IMPROVES WITH EXPERIENCE E

EXPERIENCE FEEDBACK LOOP

LEARNING, VALIDATING

DATA

ML TASK

RESULTS

PREPARED, INPUT FOR

WITH PERFORMANCE

ML TASK

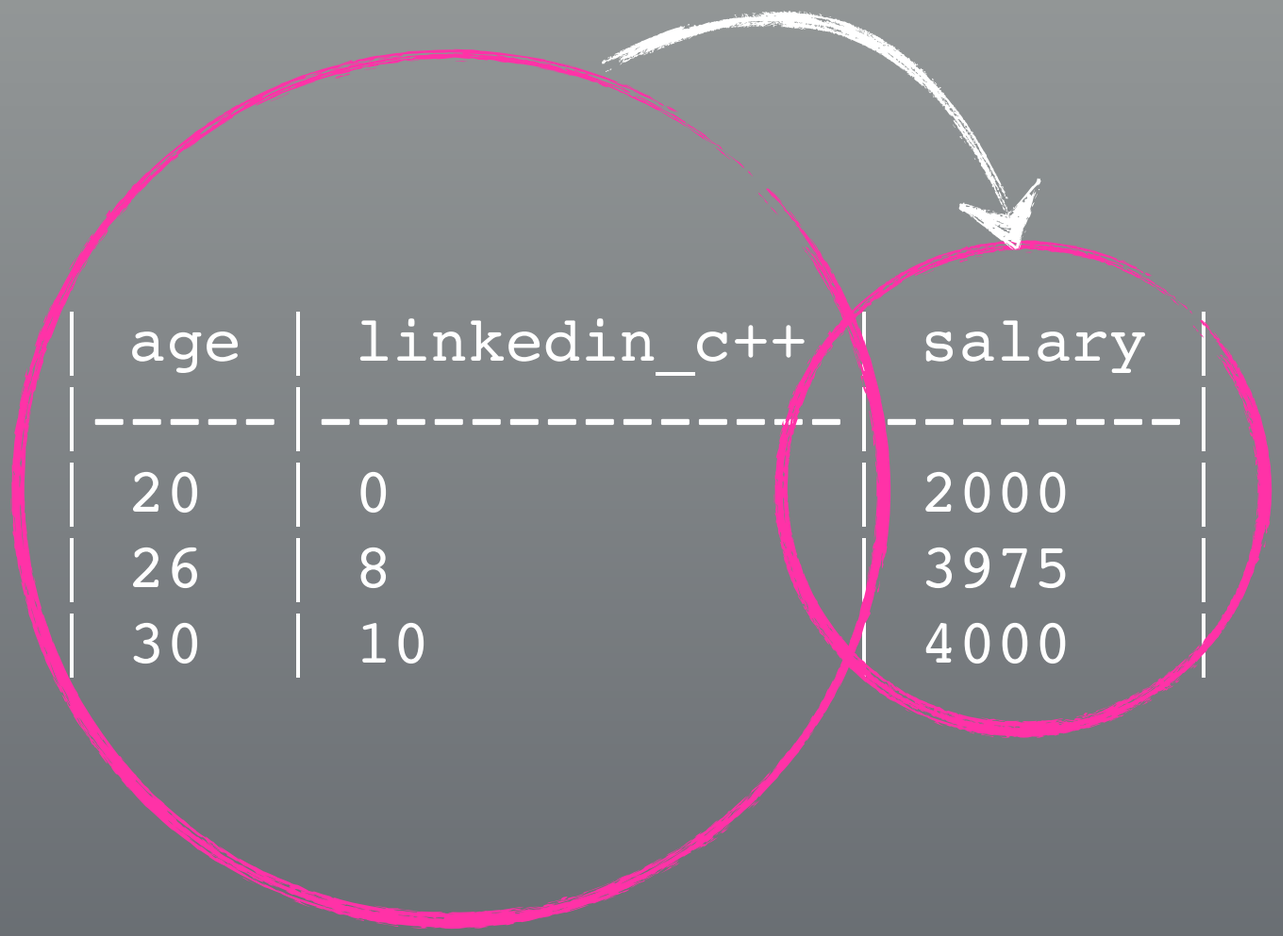
CLASSIFICATION
REGRESSION
CLUSTERING
DIMENSIONALITY REDUCTION
ASSOCIATION RULES

ML METHOD

SUPERVISED LEARNING
UNSUPERVISED LEARNING
REINFORCEMENT LEARNING

OUR PROBLEM

DEVELOPERS DATASET



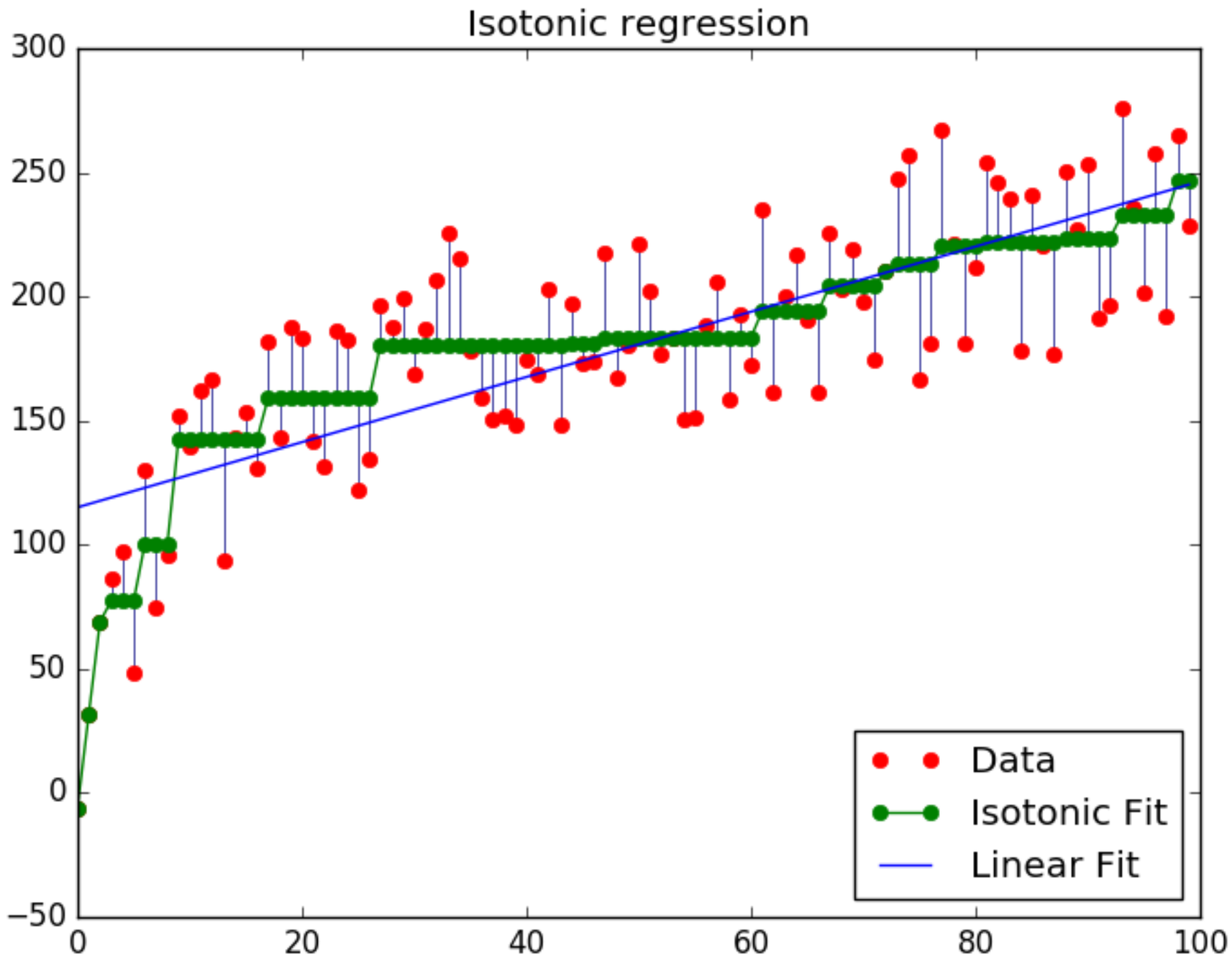
age	linkedin_c++	salary
20	0	2000
26	8	3975
30	10	4000

REGRESSION

PREDICTING VALUES

HOW MANY ~~**BRITISH POUNDS**~~ . . . **EURO**
I SHOULD **EARN** AS DEVELOPER
ACCORDING TO MY **SKILLSET**?

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection

from sklearn.linear_model import LinearRegression
from sklearn.isotonic import IsotonicRegression
from sklearn.utils import check_random_state

n = 100
x = np.arange(n)
rs = check_random_state(0)
y = rs.randint(-50, 50, size=(n,)) + 50. * np.log(1 + np.arange(n))

ir = IsotonicRegression()
y_ = ir.fit_transform(x, y)

lr = LinearRegression()
lr.fit(x[:, np.newaxis], y)
```


Isotonic regression
300
250
200
150
100
50
0
−50
0
20
40
60
80
100
Data
Isotonic Fit
Linear Fit

DON'T YOU KNOW YOUR DATA?

RECIPE FOR A FAILURE

WOKRKING ON ML, APPLY

UDF*

age	linkedin_c++	salary
20	0	2000
26	8	3975
30	10	4000

age	city_size	linkedin_c++	salary
20	900000	0	2000
20	400000	0	1800
25	450000	8	3700
26	900000	8	3975
30	100000	10	4000
30	500000	10	3500

CLUSTERING

K-MEANS

HOW TO **CLUSTER** OUR DATASET **AUTOMATED WAY** IN THE CONTEXT OF...?

1936, RONALD FISHER

IRIS DATASET

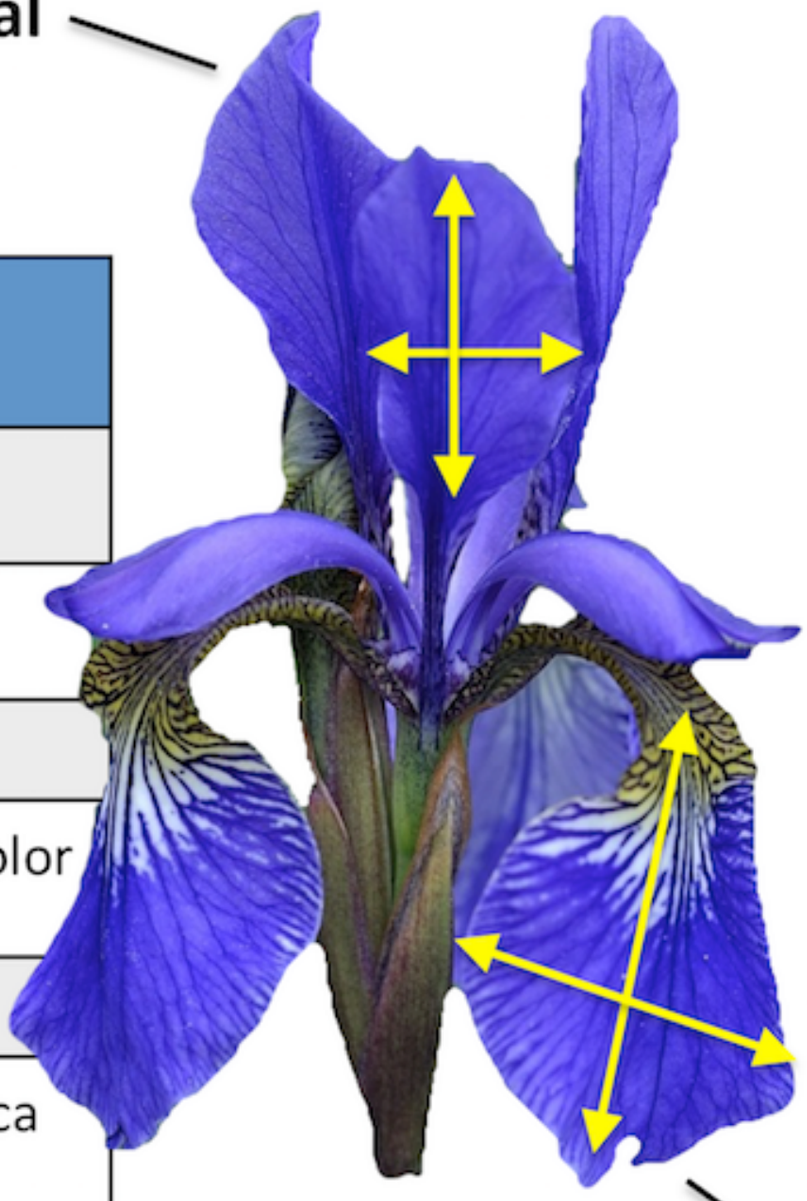
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

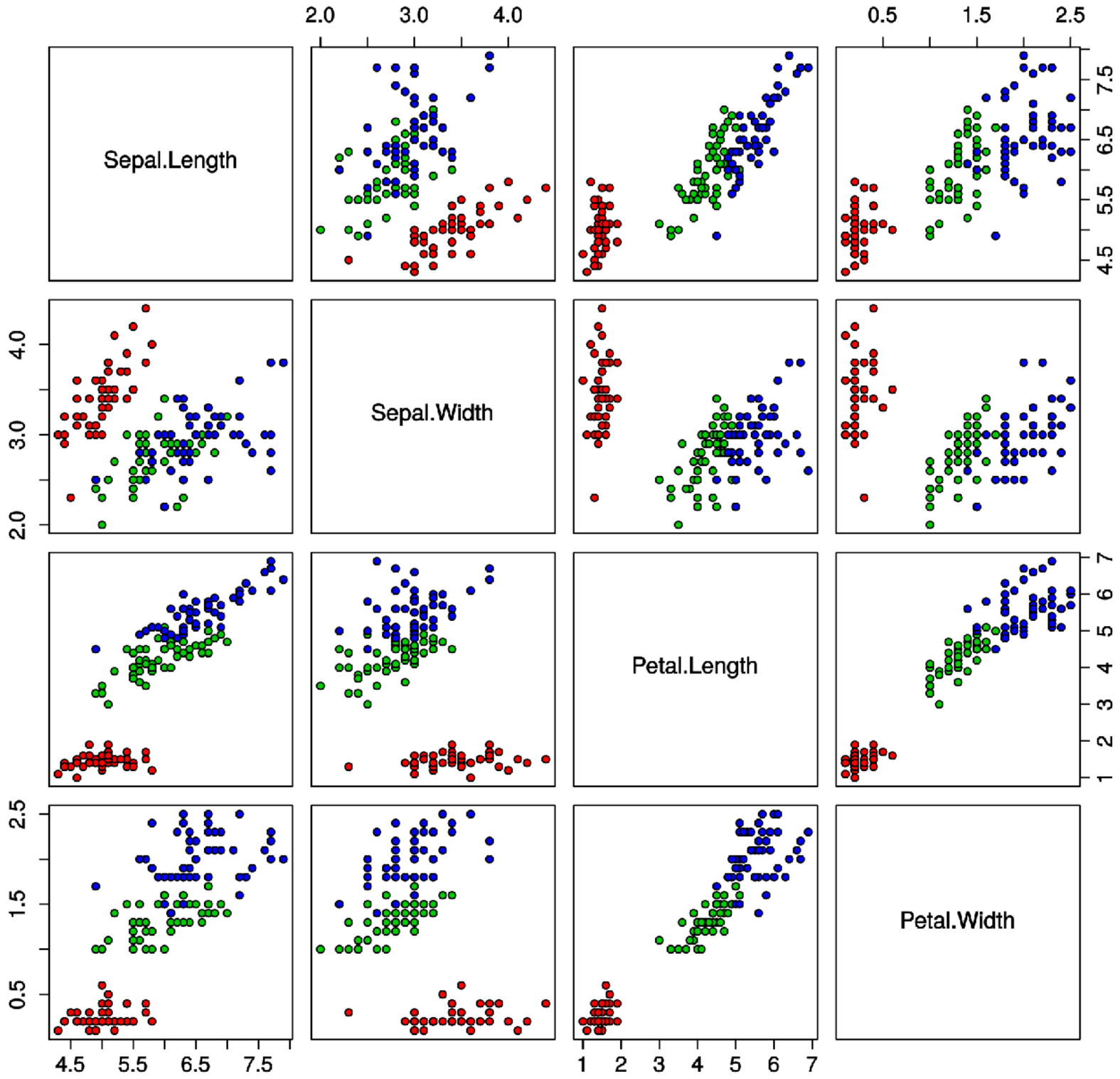
Features
(attributes, measuremer

Class labels
(targets)

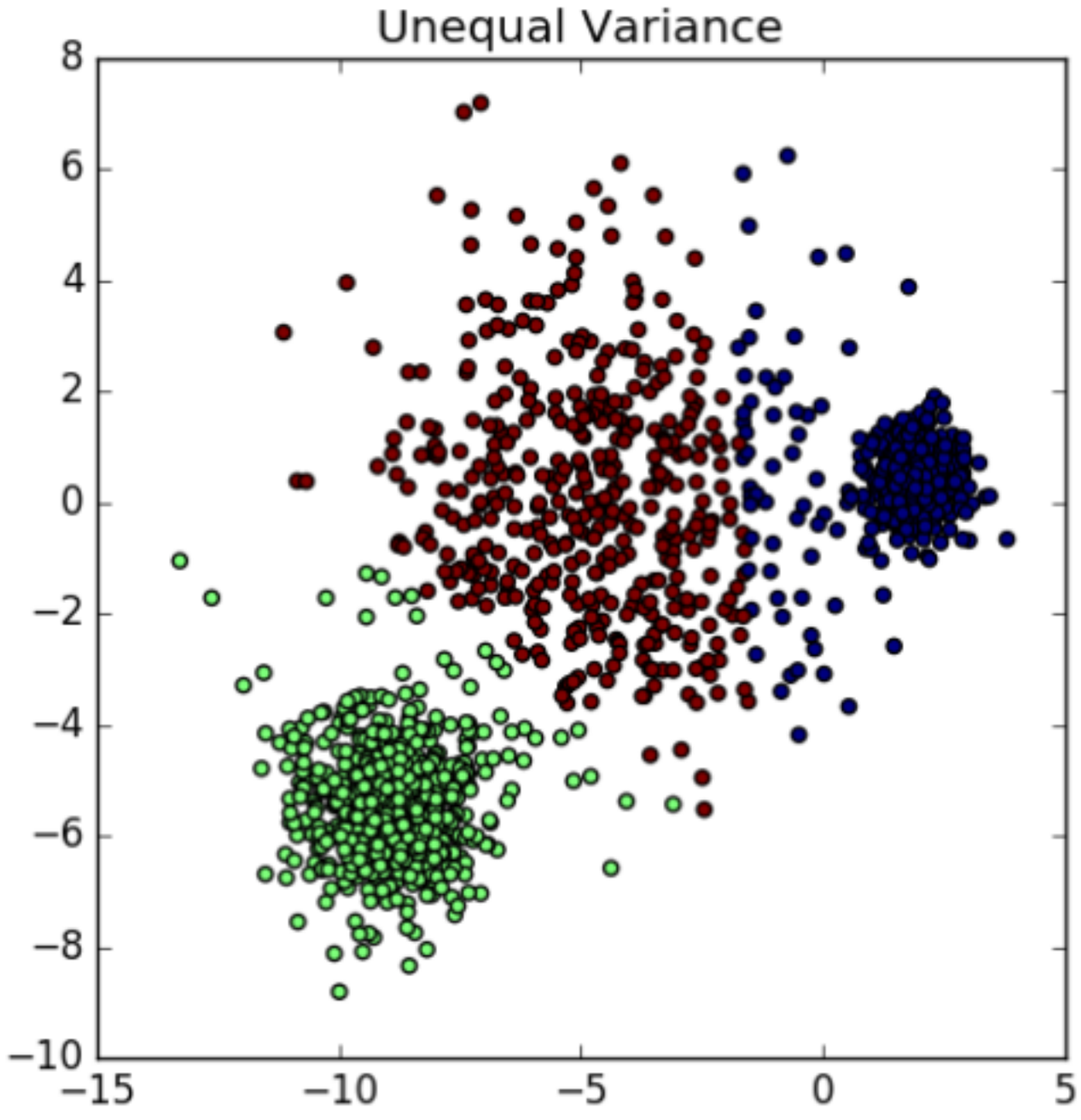
Ronald Fisher
1936 paper
The use of multiple measurements in taxonomic problems



Iris Data (red=setosa,green=versicolor,blue=virginica)



```
import numpy as np
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

n_samples = 1500
random_state = 170
X, y = make_blobs(n_samples=n_samples, random_state=random_state)

X_varied, y_varied = make_blobs(n_samples=n_samples,
                                cluster_std=[1.0, 2.5, 0.5],
                                random_state=random_state)
y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X_varied)
```

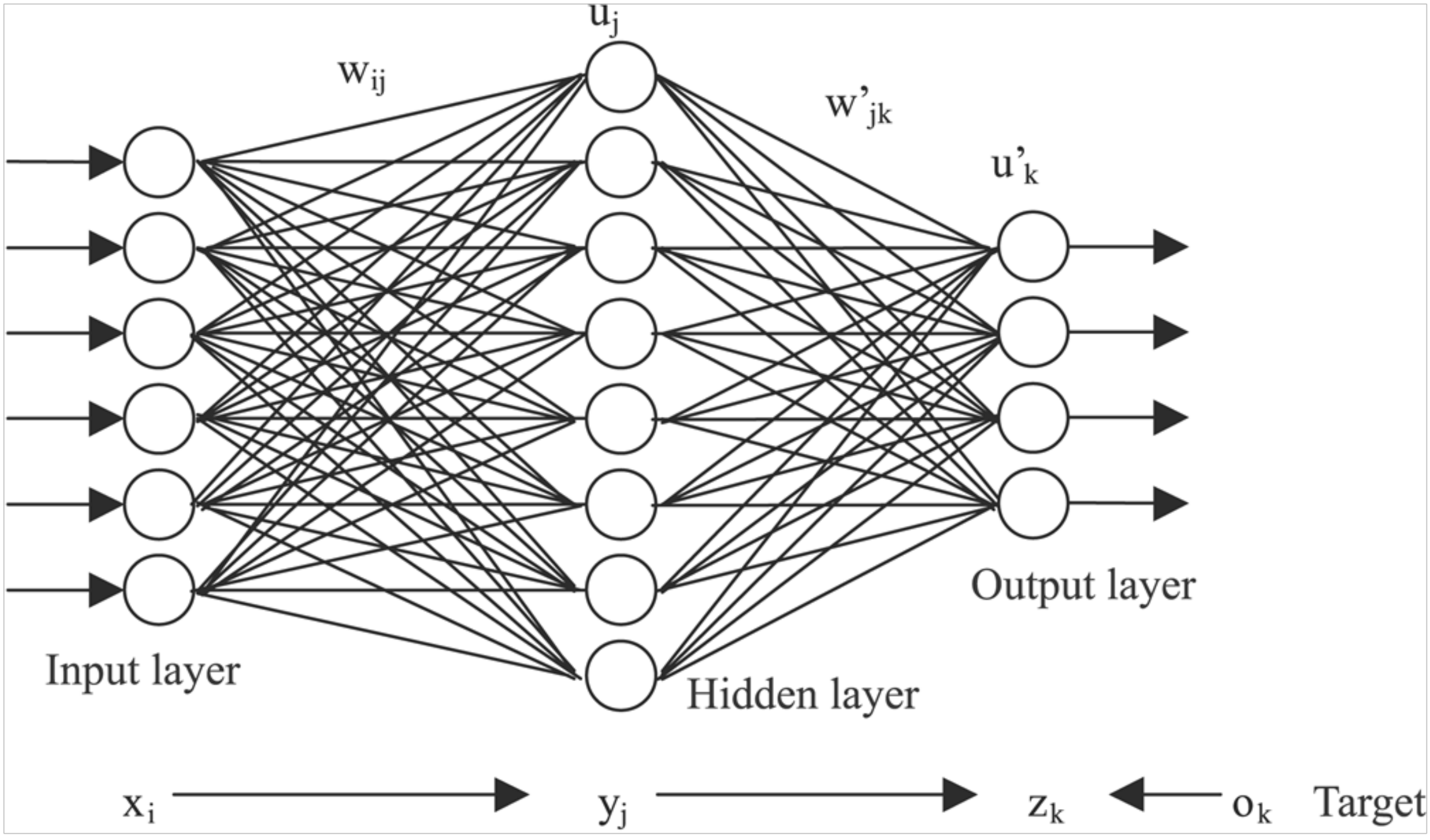

Unequal Variance
8
6
4
2
0
−2
−4
−6
−8
−10
−15
−10
−5
0
5

RESULTS STABILITY

CLASSIFICATION

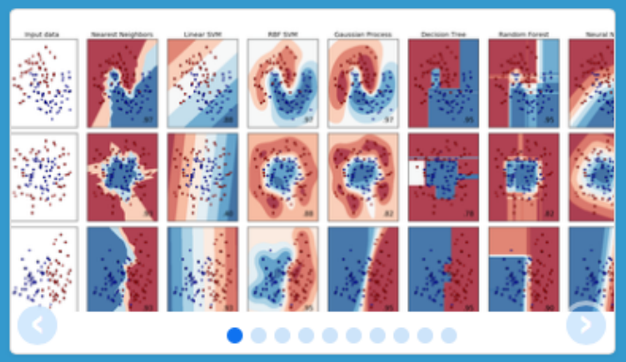
FAST ARTIFICIAL NEURAL NETWORK

HOW TO **CLASSIFY** OUR DATASET **AUTOMATED WAY** TO FIND JUNIOR/SENIOR DEVELOPER?


u_j
w_{ij}
w'_{jk}
u'_k
Input layer
Hidden layer
Output layer
x_i
y_j
z_k
o_k
Target

```
from sklearn.datasets import fetch_mldata
from sklearn.neural_network import MLPClassifier

mnist = fetch_mldata("MNIST original")

# rescale the data, use the traditional train/test split
X, y = mnist.data / 255., mnist.target
X_train, X_test = X[:60000], X[60000:]
y_train, y_test = y[:60000], y[60000:]

# mlp = MLPClassifier(hidden_layer_sizes=(100, 100), max_iter=400, alpha=1e-4,
#                     solver='sgd', verbose=10, tol=1e-4, random_state=1)
mlp = MLPClassifier(hidden_layer_sizes=(50,), max_iter=10, alpha=1e-4,
                    solver='sgd', verbose=10, tol=1e-4, random_state=1,
                    learning_rate_init=.1)

mlp.fit(X_train, y_train)
```

TECHNOLOGY



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

News

On-going development: What's new (Changelog)

November 2016. scikit-learn 0.18.1 is available for download (Changelog).

September 2016. scikit-learn 0.18.0 is available for download (Changelog).

November 2015. scikit-learn 0.17.0 is available

Community

About us See authors and contributing

More Machine Learning Find related projects

Questions? See FAQ and stackoverflow

Mailing list: scikit-learn@python.org

IRC: #scikit-learn @ freenode

Who uses scikit-learn?

TELECOM ParisTech

excellent scikit- "The great benefit of scikit-learn is
ning curve [...]"

More testimonials

mlpack

main about docs get help github

mlpack is a scalable machine learning library, written in C++, that aims to provide fast, extensible implementations of cutting-edge machine learning algorithms. **mlpack** provides these algorithms as simple command-line programs and C++ classes which can then be integrated into larger-scale machine learning solutions.

get mlpack

- download (newest version **2.1.0**, released **oct. 31st, 2016**)
- release history
- citation information

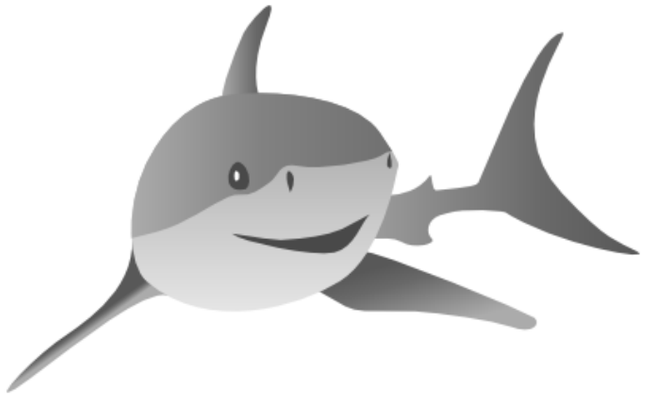
learn about mlpack

- what is mlpack and what does it implement?
- how fast is mlpack?
- how do I get help?
- how can I join the mlpack community?
- mlpack development blog

more information

- api documentation -- 2.1.0 -- git master
- list of tutorials -- 2.1.0 -- git master
- command-line program reference -- 2.1.0 -- git master
- documentation for all versions
- papers: BigLearning, JMLR
- site map

Shark – Machine Learning 3.1



News

Shark 3.1 Released

March 1, 2016: We are happy to announce the official release of Shark 3.1.0.

Shark 3.0 Released

October 27, 2015: We are happy to announce the official release of Shark 3.0.0.

Shark moves to GitHub

October 9, 2015: Shark moved to GitHub. Please update your repositories, see the downloads page for more details.

Shark goes LGPL

As of *January 2014*, Shark is distributed under the permissive GNU Lesser General Public License.

deeplearning4j.org
DEEPLEARNING4J
QUICKSTART
DOCUMENTATION
GPUS
SPARK
WORD2VEC
ABOUT
Fork me on GitHub
Deep Learning for Java
Open-Source, Distributed, Deep Learning Library for the JVM
GET STARTED
Star 4,519
Fork 1,524
Watch 473
OPEN CHAT

spark.apache.org/mllib/

Apache Spark MLlib

Download | Libraries | Documentation | Examples | Community | FAQ | Apache Software Foundation

MLlib is Apache Spark's scalable machine learning library.

Ease of Use

Usable in Java, Scala, Python, and R.

MLlib fits into Spark's APIs and interoperates with NumPy in Python (as of Spark 0.9) and R libraries (as of Spark 1.5). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

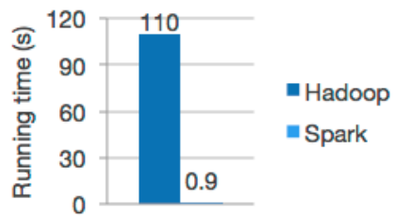
```
data = spark.read.format("libsvm")\
  .load("hdfs://...")

model = KMeans(data, k=10)
```

Calling MLlib in Python

Performance

High-quality algorithms, 100x faster than MapReduce.

Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.



Logistic regression in Hadoop and Spark

Easy to Deploy

Runs on existing Hadoop clusters and data.

If you have a Hadoop 2 cluster, you can run Spark and MLlib without any pre-installation. Otherwise, Spark is easy to run standalone or on EC2 or Mesos. You can read from HDFS, HBase, or any Hadoop data source.

Latest News

Spark 1.6.3 released (Nov 07, 2016)
Spark 2.0.1 released (Oct 03, 2016)
Spark 2.0.0 released (Jul 26, 2016)
Spark 1.6.2 released (Jun 25, 2016)

Archive

Download Spark

Built-in Libraries:

SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Packages

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

Documentation

Manuals
FAQs
The R Journal
Books
Certification
Other

Links

Bioconductor
Related Projects

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **R version 3.3.2 (Sincere Pumpkin Patch)** has been released on Monday 2016-10-31.
- **The R Journal Volume 8/1** is available.
- The **useR! 2017** conference will take place in Brussels, July 4 - 7, 2017, and details will be appear here in due course.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, hase taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

1 Introduction
2 Visualizations
3 Pre-Processing
4 Data Splitting
5 Model Training and Tuning
6 Available Models
7 train Models By Tag
8 Models Clustered by Tag Similarity
9 Parallel Processing
10 Random Hyperparameter Search
11 Subsampling For Class Imbalances
12 Using Your Own Model in train
13 Adaptive Resampling
14 Variable Importance
15 Miscellaneous Model Functions
16 Measuring Performance
17 Feature Selection Overview
18 Recursive Feature Elimination
19 Feature Selection using Univariate F...
20 Feature Selection using Genetic Alg...
21 Feature Selection using Simulated A...
22 Data Sets
23 Session Information

The `caret` Package

Max Kuhn

2016-11-10

1 Introduction

The `caret` package (short for *C*lassification *A*nd *RE*gression *T*raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

as well as other functionality.

There are many different modeling functions in R. Some have different syntax for model training and/or prediction. The package started off as a way to provide a uniform interface the functions themselves, as well as a way to standardize common tasks (such parameter tuning and variable importance).

The current release version can be found on CRAN and the project is hosted on github.

Some resources:

- The book *Applied Predictive Modeling* features **caret** and over 40 other R packages. It is on sale at Amazon or the the publisher's website. There is a companion website too.
- There is also a paper on caret in the Journal of Statistical Software. The example data can be obtained here(the predictors) and here (the outcomes).
- There is a webinar for the package on Youtube that was organized and recorded by Ray DiGiacomo

shiny.rstudio.com

Fork me on GitHub

Shiny

by RStudio

A web application framework for R

Turn your analyses into interactive web applications

No HTML, CSS, or JavaScript knowledge required

TUTORIAL ARTICLES GALLERY REFERENCE DEPLOY HELP

Get inspired
(gallery)

Get started
(tutorial)

Go deeper
(articles)

Here is a Shiny app

Shiny apps are easy to write. No web development skills are required.

Number of bins in histogram (approximate):

20

Show individual observations

Show density estimate

Geyser eruption duration

ui.R server.R

```
shinyUI(bootstrapPage(

  selectInput(inputId = "n_breaks",
      label = "Number of bins in histogram (approximate):",
      choices = c(10, 20, 35, 50),
      selected = 20),

  checkboxInput(inputId = "individual_obs",
      label = strong("Show individual observations"),
      value = FALSE),
```

rApache

downloads

Download rApache 1.2.8

To follow the latest development on github.com, go here:

https://github.com/jeffreyhorner/rapache

View the Changelog

https://raw.github.com/jeffreyhorner/rapache/master/ChangeLog

Ubuntu Package

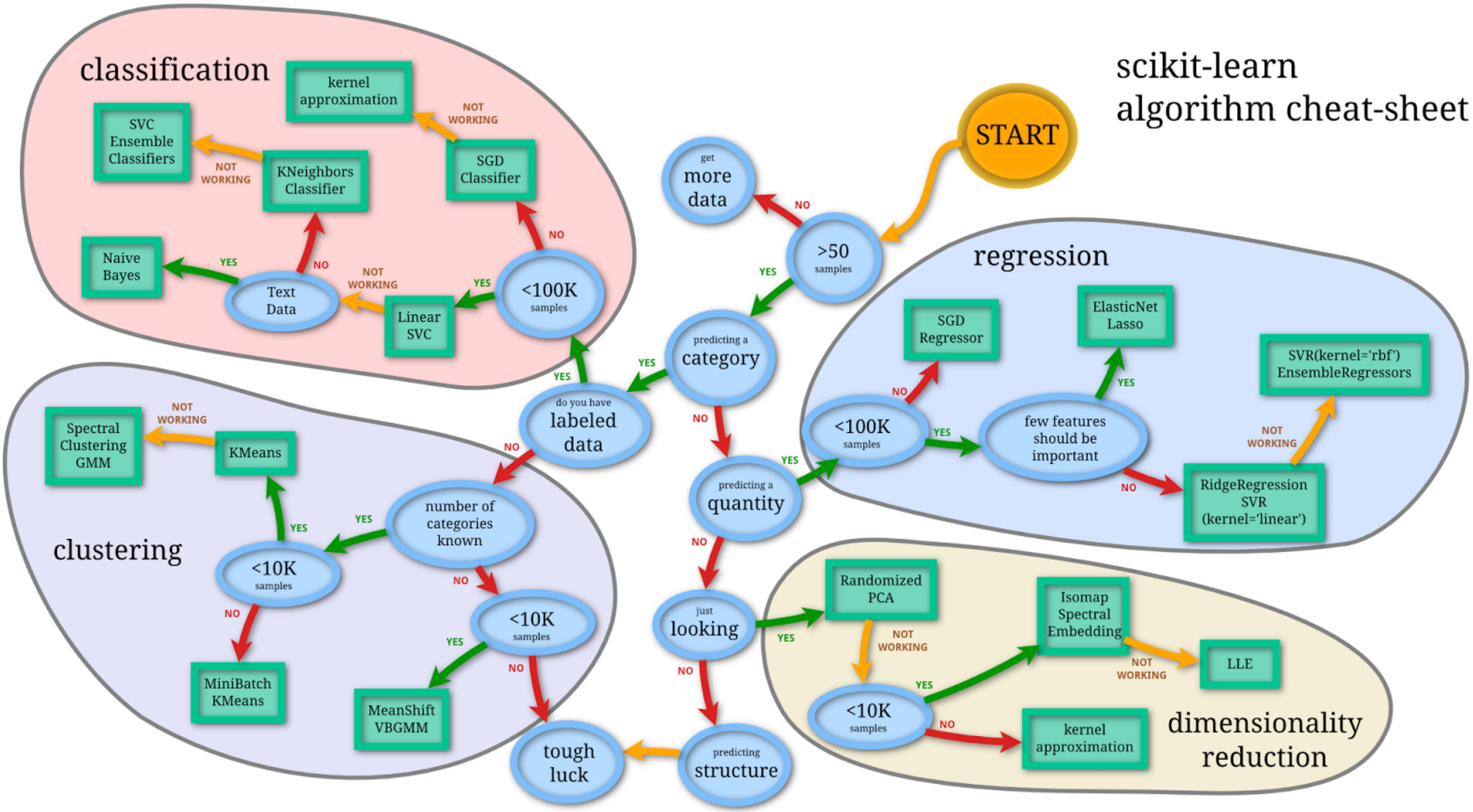
Convenient installation packages for Ubuntu 12.04 (Precise) and newer are available through Launchpad:

```
sudo add-apt-repository ppa:opencpu/rapache
sudo apt-get update
sudo apt-get install libapache2-mod-r-base
```

© 2013 Vanderbilt University | License (Apache 2.0) | Styled with skeleton.

FOCUS ON IDEAS NOT TOOLS

scikit-learn algorithm cheat-sheet



http://scikit-learn.org/stable/_static/ml_map.png

ML IS NOT A SINGLE RUN OF ALGORITHM

IT'S A PROCESS

ML PROCESS

DEFINE A PROBLEM
ANALYZE YOUR DATA
UNDERSTAND YOUR DATA
PREPARE DATA FOR ML
SELECT & RUN ALGO(S)
TUNE ALGO(S) PARAMETERS
SELECT FINAL MODEL
VALIDATE FINAL MODEL

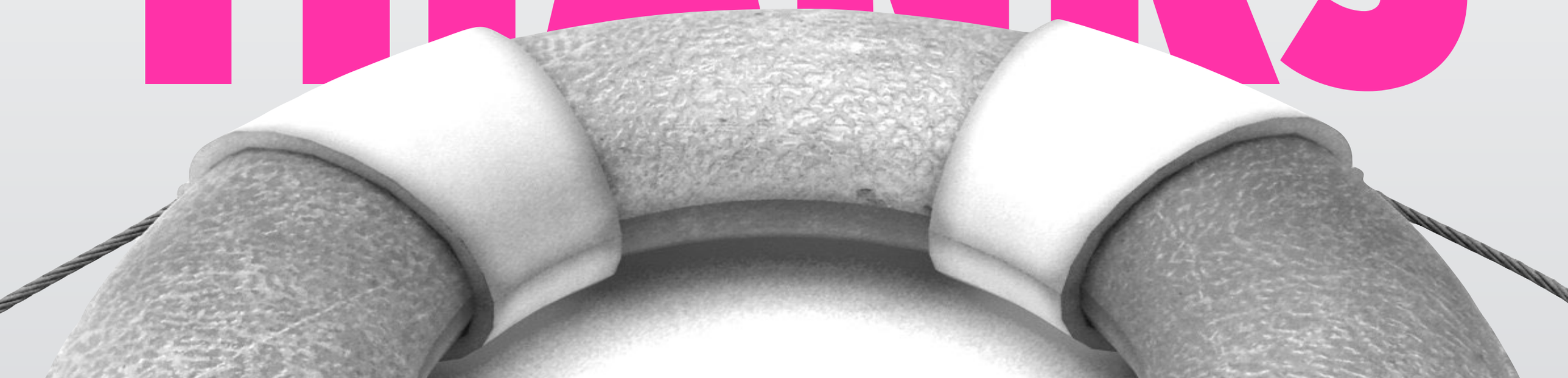
ML PROCESS

DEFINE A PROBLEM
ANALYZE YOUR DATA
UNDERSTAND YOUR DATA
PREPARE DATA FOR ML
SELECT & RUN ALGO(S)
TUNE ALGO(S) PARAMETERS
SELECT FINAL MODEL
VALIDATE FINAL MODEL

age	city_size	linkedin_c++	salary
20	900000	0	2000
20	400000	0	1800
25	450000	8	3700
26	900000	8	3975
30	100000	10	4000
30	500000	10	3500

age	city_size	linkedin_c++	salary	currency
20	900000	0	2000	EUR
20	400000	0	1800	USD
25	450000	8	3700	USD
26	900000	8	3975	USD
30	100000	10	4000	USD
30	500000	10	3500	USD

mariuszgil

HAPPY LEARNING YOUR MACHINES!

THANKS