



Data science piaci körkép

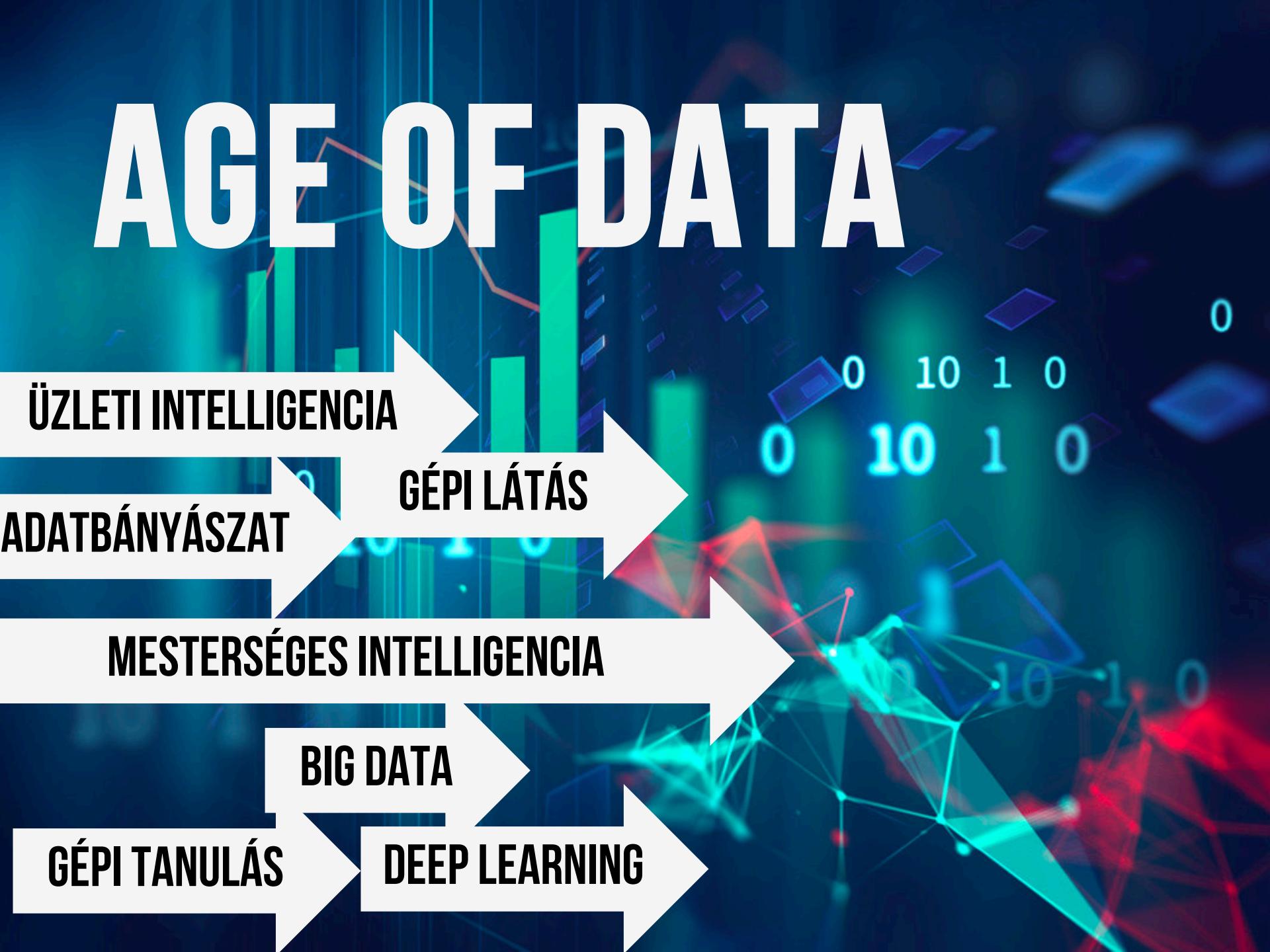
Gáspár Csaba
data scientist



AGENDA

- Big data eredetmonda
 - Miért van mindennek furcsa neve?
- Általános termékjellemzők
 - Miért nem vette már meg mindenki?
- Data scientist, mint munkakör
 - Miért találták ki az új munkakört?
- Globális trend
 - Covid és a háború hatása
- Budapest az adatok világában
- Hazai piac szerkezete

AGE OF DATA



ÜZLETI INTELLIGENCIA

ADATBÁNYÁSZAT

MESTERSÉGES INTELLIGENCIA

BIG DATA

GÉPI TANULÁS

DEEP LEARNING

Company	2010 Market Share (%)
SAP	23.0
Oracle	15.7
SAS Institute	13.2
IBM	11.6
Microsoft	8.7
Other Vendors	27.9



IBM SPSS Modeler

Categorical_Grouping* - IBM® SPSS® Modeler

The diagram illustrates a data flow process in IBM SPSS Modeler. It starts with an input node (Response) which branches into Type and Product_Group. Type further branches into Training, Response Rate, Decile, and Product_Group. Product_Group branches into No Group and With Group. No Group leads to Response NG, which then connects to Response WG. With Group also leads to Response WG. Both Response WG and Response NG lead to Analysis.

Streams Outputs Models

- Stream1
- Recipe - variable construct mu
- Categorical_Grouping

CRISP-DM Classes

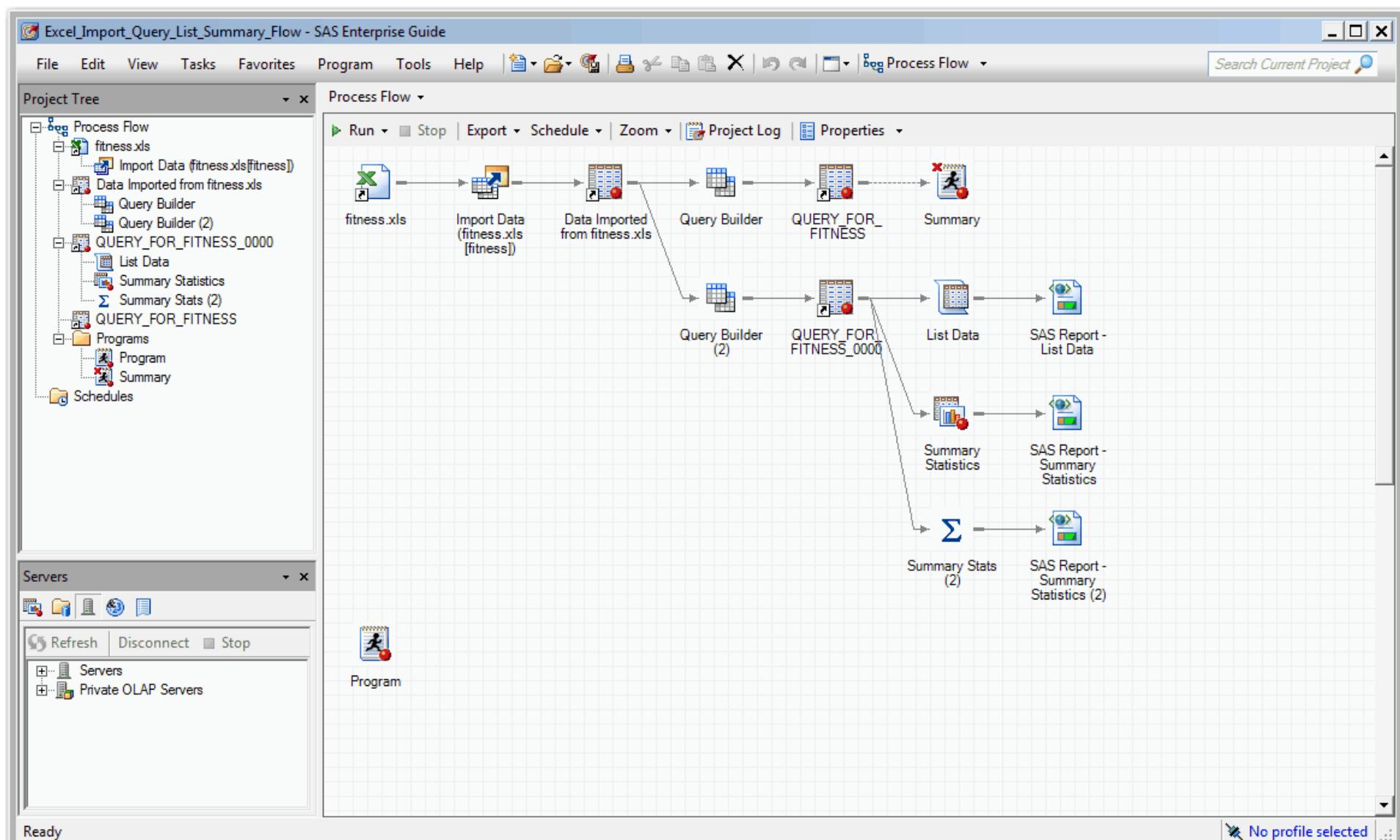
- (unsaved project)
- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment

Favorites Sources Record Ops Field Ops Graphs Modeling Output Export IBM® SPSS® Statistics Python Spark

Database Var. File Auto Data Prep Select Sample Aggregate Derive Type Filter Graphboard Auto Classifier Auto Numeric Auto Cluster Table Flat File Database

Server Local Server 123MB / 349MB

SAS Enterprise Miner

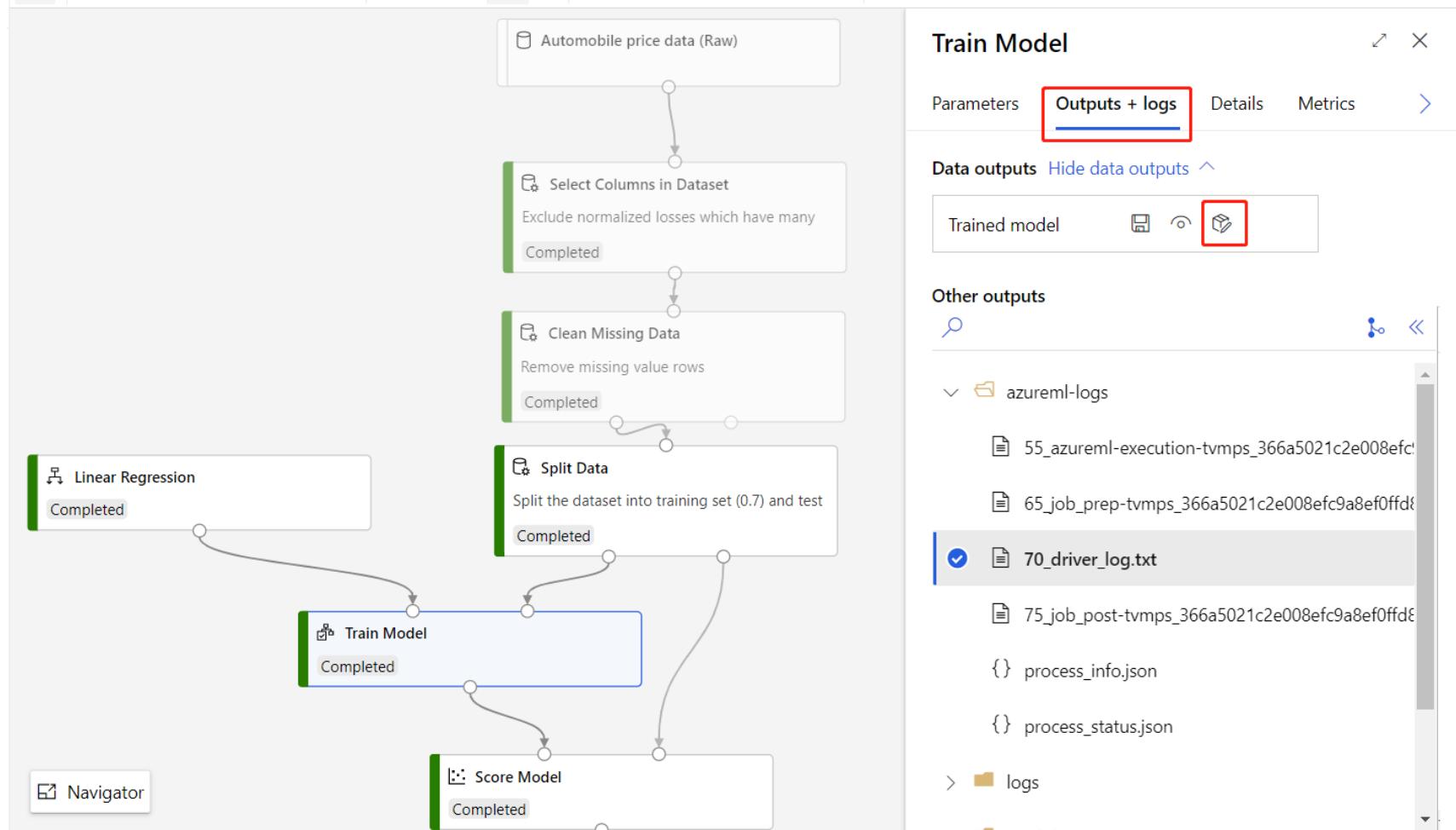


Azure Machine Learning

Regression - Automobile Price Prediction (Basic) 

Submit Create inference pipeline Publish ...

Autosave on  Run finished [View run overview](#)



The screenshot shows a completed machine learning pipeline for a regression task. The pipeline consists of the following steps:

- Automobile price data (Raw)**: The starting dataset.
- Select Columns in Dataset**: Excludes normalized losses which have many completed values.
- Clean Missing Data**: Removes missing value rows.
- Linear Regression**: A completed model step.
- Split Data**: Splits the dataset into training set (0.7) and test set.
- Train Model**: Trains the Linear Regression model on the training data.
- Score Model**: Scores the trained model on the test data.

The pipeline is visualized as a flowchart with arrows indicating the data flow between steps. The "Outputs + logs" tab in the "Train Model" panel is selected, and the "Trained model" output is highlighted with a red box. The "Outputs + logs" section also lists other outputs such as "azureml-logs" and "70_driver_log.txt".

Oracle Data Mining

Oracle SQL Developer : Oracle 12c DMUSER/4.0 New Features/Oracle Data Miner 4.0

File Edit View Navigate Run Diagram Team Tools Window Help

Connections Data Miner

Start Page Oracle Data Miner 4.0 Profile Data MINING_DATA_BUILD_TEXT2 Apply 1.sql Clustering Predictive Query Parallel Query Off

100%

Components Workflow Editor

Data Create Table or View Data Source Explore Data

Graph SQL Query Update Table

Transforms

Aggregate Filter Columns Filter Columns Details

Filter Rows Join Sample

Text

Apply Text Build Text Text Reference

Models

Anomaly Detection Association Classification

Clustering Feature Extraction Model

Predictive Queries

Anomaly Detection Query Clustering Query Feature Extraction Query

Prediction Evaluate and Apply Linking Nodes

Parallel Query Off

Profile Data

Clustering Segmentation

NEW CUST_INSUR_LTV_APPLY

Filter Cols_Attrib Importance

Predictive Models for Buyers

Likely Buyers

Model Details

Filter Columns Details

Predictive Query

Connect

Edit...

Validate Parents

Run

Force Run

View Data

Save SQL

Deploy

Generate Apply Chain

Cut Ctrl+X

Copy Ctrl+C

Paste Ctrl+V

Extended Paste... Ctrl+Shift+V

Select All Ctrl+A

Anomaly Detection Reports

Data Dictionary Reports

OLAP Reports

TimesTen Reports

User Defined Reports

Reports

LM Build Model

INSTALL_R_PACKAGES5

Regress Build

Filter Columns Details

Model Details

Model Details 1

Join

Predictive Models for Buyers

Predictive Models

Reports

All Reports

Parallel Query ...

Copy Image to Clipboard

Save Image As...

Show Event Log

Go to Properties

Navigate

Column PRED_1

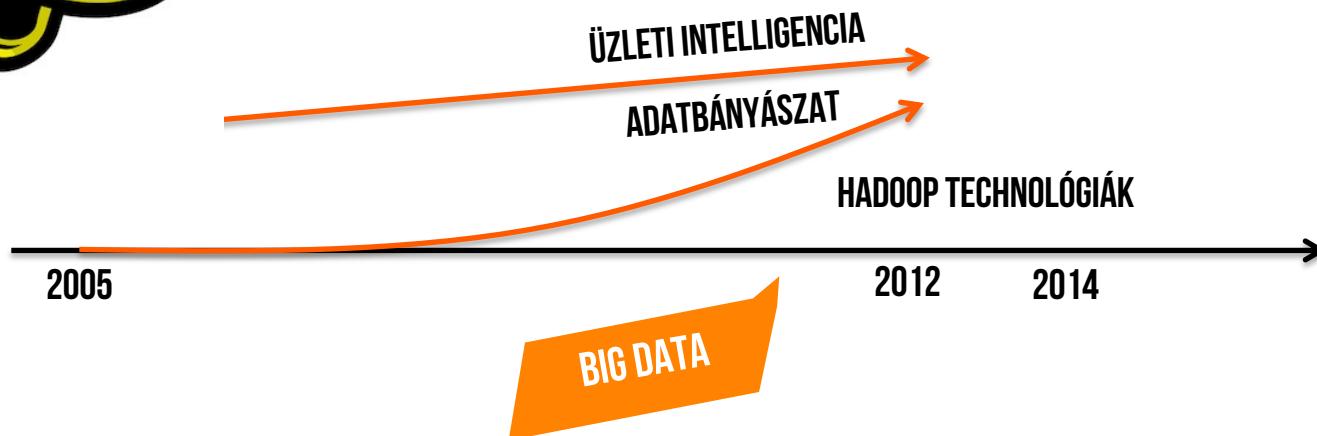
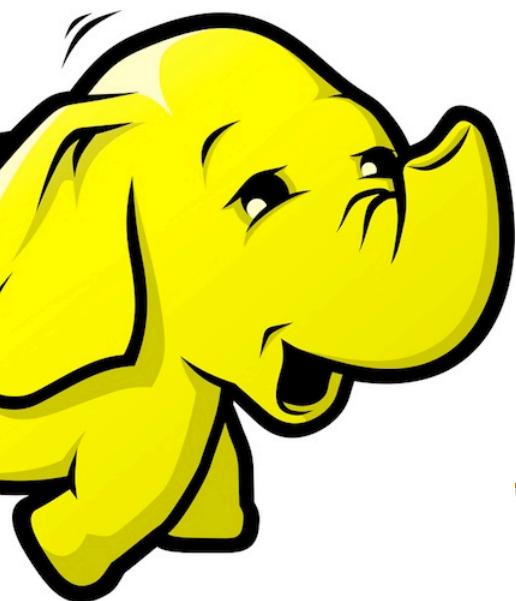
Parameter

Opened nodes (106); Saved files(0)

4:46 PM 4/29/2014

The screenshot shows the Oracle Data Miner 4.0 interface within Oracle SQL Developer. The main workspace displays a workflow diagram consisting of several nodes: 'Profile Data', 'Clustering Segmentation', 'Filter Cols_Attrib Importance', 'Predictive Models for Buyers', 'Likely Buyers', and 'Model Details'. Arrows indicate the flow of data between these nodes. A context menu is open over the 'Profile Data' node, listing options like 'Edit...', 'Validate Parents', 'Run', 'Force Run', 'View Data', 'Save SQL', 'Deploy', 'Generate Apply Chain', and various cut/copy/paste commands. The 'Components' palette on the right side lists various data mining components such as 'Create Table or View', 'Data Source', 'Explore Data', 'Graph', 'SQL Query', 'Update Table', 'Aggregate', 'Filter Columns', 'Filter Rows', 'Join', 'Sample', 'Text', 'Apply Text', 'Build Text', and 'Text Reference'. The 'Models' section includes 'Anomaly Detection', 'Association', 'Classification', 'Clustering', 'Feature Extraction', and 'Model'. The 'Predictive Queries' section lists 'Anomaly Detection Query', 'Clustering Query', 'Feature Extraction Query', 'Prediction', 'Evaluate and Apply', and 'Linking Nodes'. The bottom status bar indicates 'Opened nodes (106); Saved files(0)' and the system time '4:46 PM 4/29/2014'.

BIG DATA EREDETMONDA



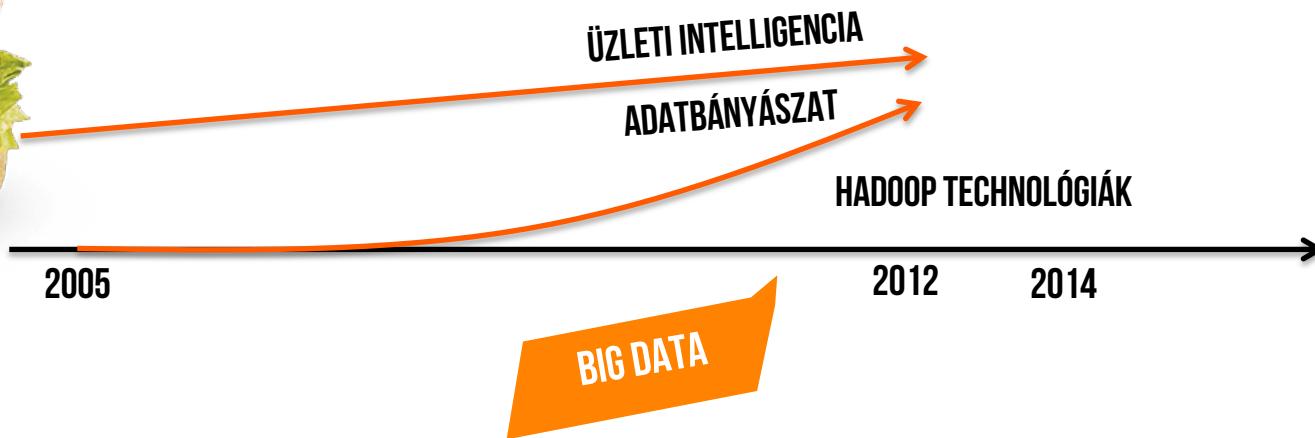
2005

2012

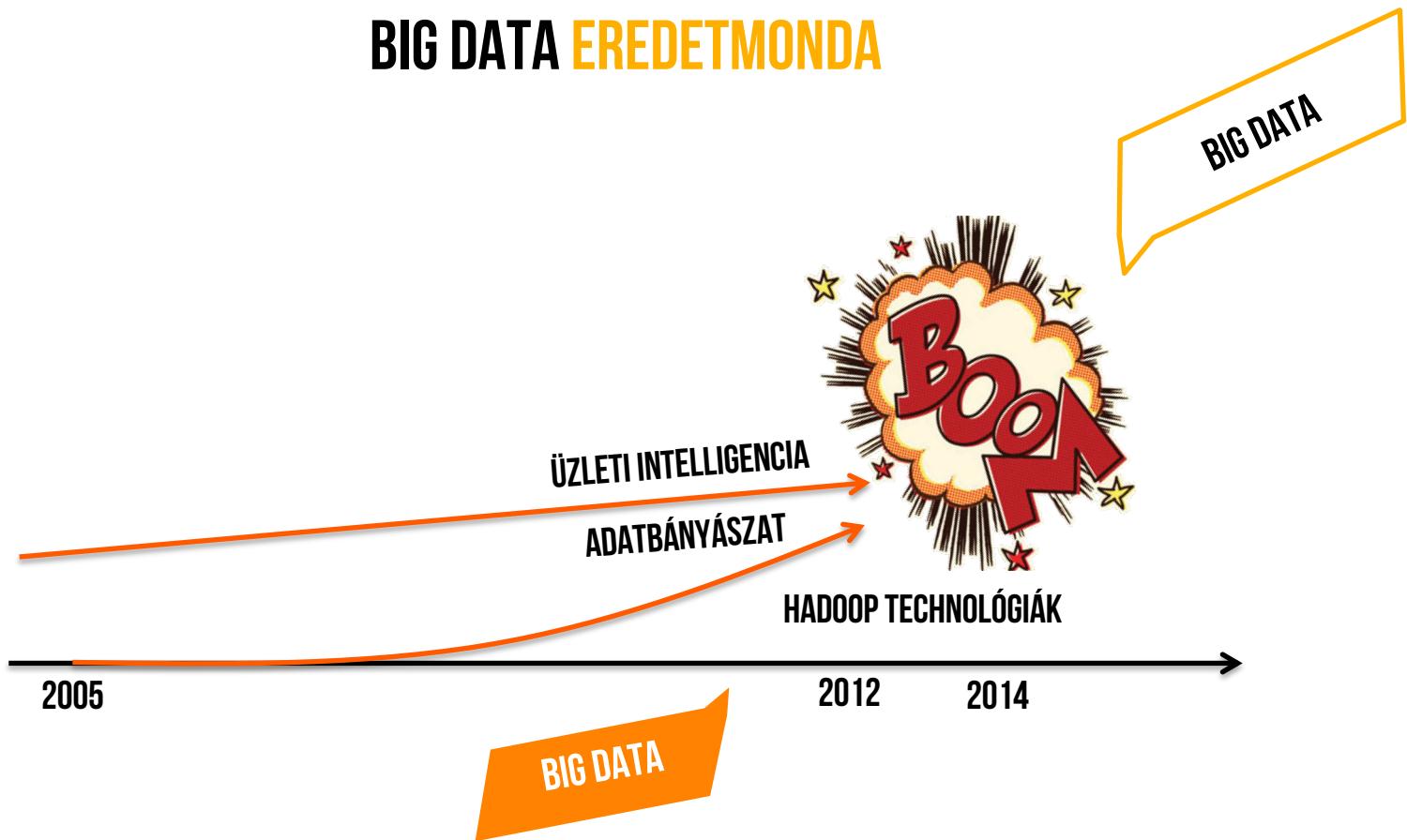
2014

BIG DATA

BIG DATA EREDETMONDA



BIG DATA EREDETMONDA



score

120

GOOGLE TRENDS NEWS SEARCH

100

80

60

40

20

0

2006

2007

2008

2009

2010

2011

2012

2013

2014

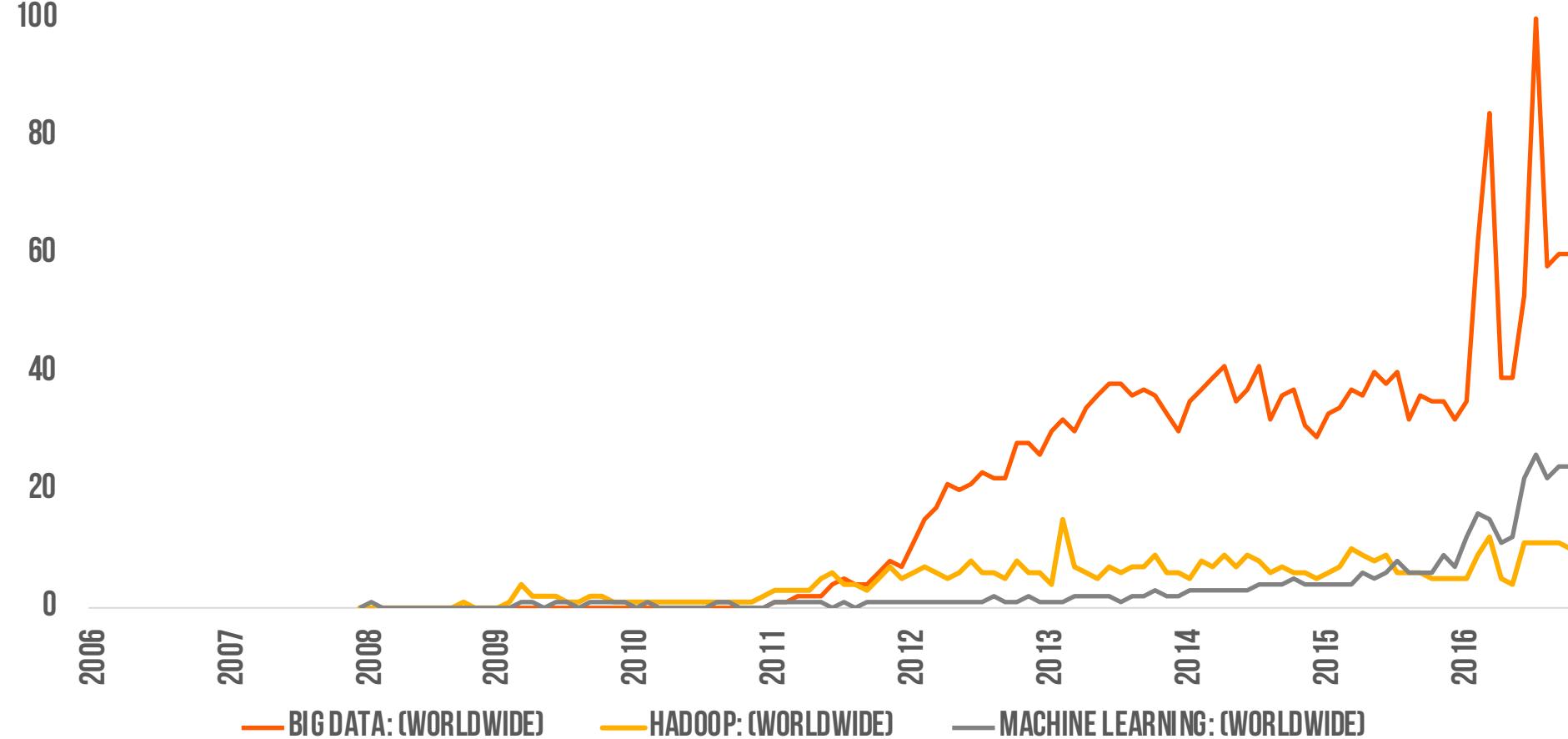
2015

2016

— BIG DATA: (WORLDWIDE)

— HADOOP: (WORLDWIDE)

— MACHINE LEARNING: (WORLDWIDE)



score

60

50

40

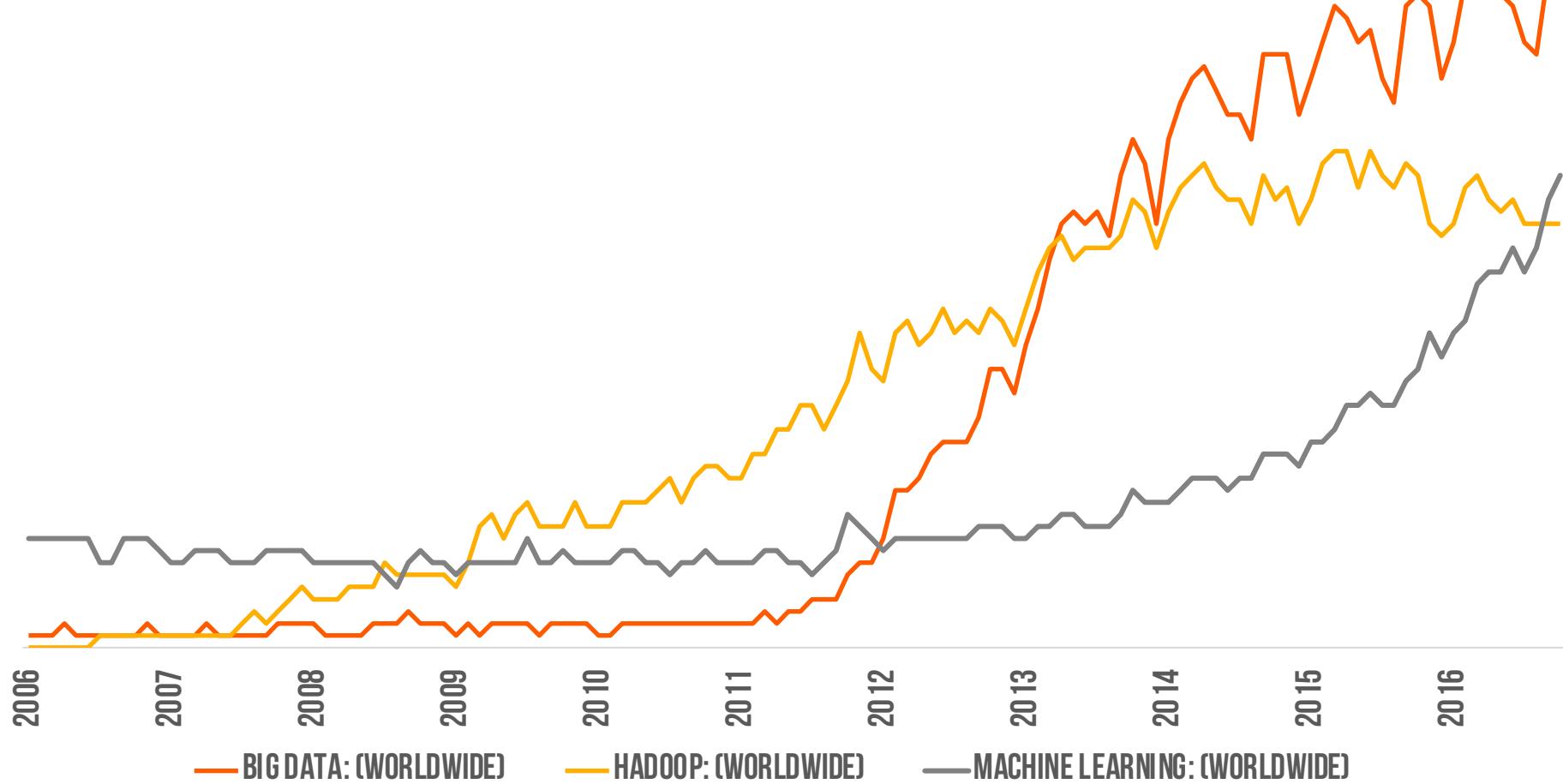
30

20

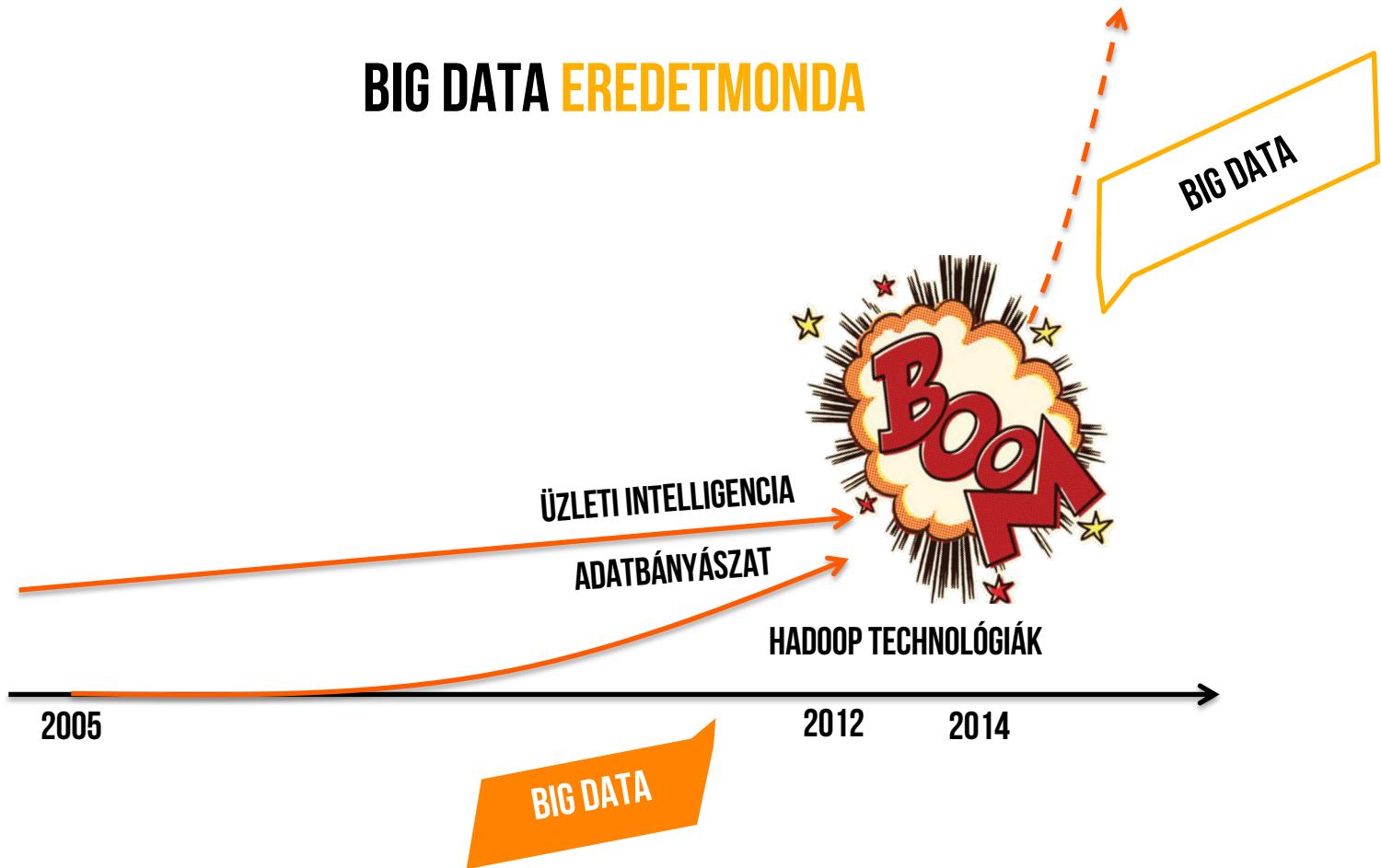
10

0

GOOGLE TRENDS WEB SEARCH



BIG DATA EREDETMONDA



BIG DATA

Technológiai

ÚJ ADATTÁROLÁSI
ÉS -FELDOLGOZÁSI PARADIGMA



- NYÍLT FORRÁSKÓD, EXTRA NAGY ADATMENNYISÉG
- OLCSÓBB PLATFORM – "CHEAP DATA"

Technológiai

BIG DATA

Big Data Landscape

Log Data Apps



Vertical Apps



Business Intelligence



Analytics and Visualization



Data Providers



Analytics Infrastructure



Operational Infrastructure



Infrastructure As A Service



Structured Databases



Technologies



- NYÍLT FORUM ADATMENNYISÉG
- OLCSÓBB PLATFORM – "CHEAP DATA"

Copyright © 2012 Dave Fiehrer

dave@vcdave.com

<http://blogs.forbes.com/davefiehr/>

2012

Technológiai

BIG
DATA

Big Data Landscape

Vertical Apps



MYRRIX

Log Data Apps



loggly



Ad/Media Apps



bluefin



Recorded Future



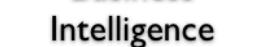
DataXU

Data, insight, Action.



TURN

Business Intelligence



SAP Business Objects

RJMetrics



Business Intelligence



COGNOS



Autonomy



qlikView



DOMO



GoodData

Analytics and Visualization



+ tableau



metaLayer



TERADAT



Sas



KARMASPHERE



Real-time Visual Data Analysis



platfora



ClearStory



visual.ly



GNIP



Windows Azure Marketplace



LexisNexis



LOQATE



knoema



Chart.io



qlikView



DOMO



cloudera



GREENPLUM



DATASTAX



calont



INFOBRIGHT



TERRACOTTA



HADAPT



INFORMATICA



TERADATA



MarkLogic



MongoDB



INFORMATICA

Infrastructure As A Service



amazon web services



infochimps



Structured Databases



Microsoft SQL Server



DB2



SYBASE



PostgreSQL

Technologies



- NYÍLT F
ADATMÉRŐ
- OLCSÓBÉRŐ PLATFORM — CHEAP DATA

Copyright © 2012 Dave Feinleib

dave@vcdave.com

blogs.forbes.com/davefeinleib

Gáspár Csaba @dmlab



ÉS - FI



• UTÁLTATÁSA

Gáspár Csaba @dmlab

2015

BIG

BIG DATA LANDSCAPE, VERSION 3.0

Exited: Acquisition or IPO

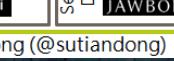
Infrastructure



Analytics



Unstructured Data



Applications



Open Source



Data Sources



© Matt Turck (@mattturck), Sutian Dong (@sutiandong) & FirstMark Capital (@firstmarkcap)

Gáspár Csaba @dmlab



Big Data Landscape 2016

Infrastructure



Cross-Infrastructure/Analytics

amazon Google Microsoft IBM SAP SAS hp VMware talend TIBCO TERADATA ORACLE NetApp

Framework



Query / Data Flow



Data Access



Coordination



Real-Time



Stat Tools



Machine Learning



Search



Security



Data Sources & APIs



FIRSTMARK

© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

Gáspár Csaba @dmlab

2017

BIG

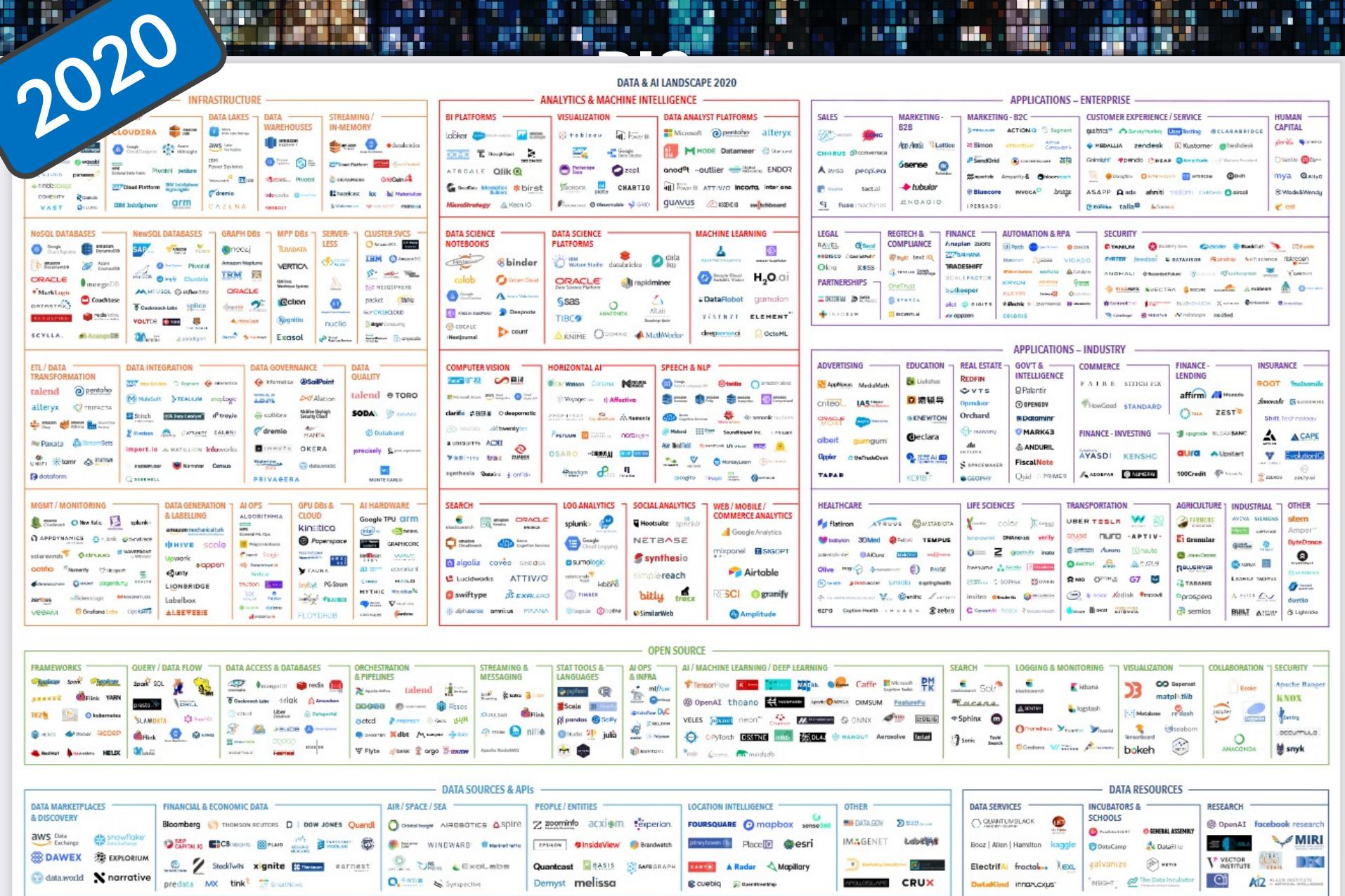
BIG DATA LANDSCAPE 2017



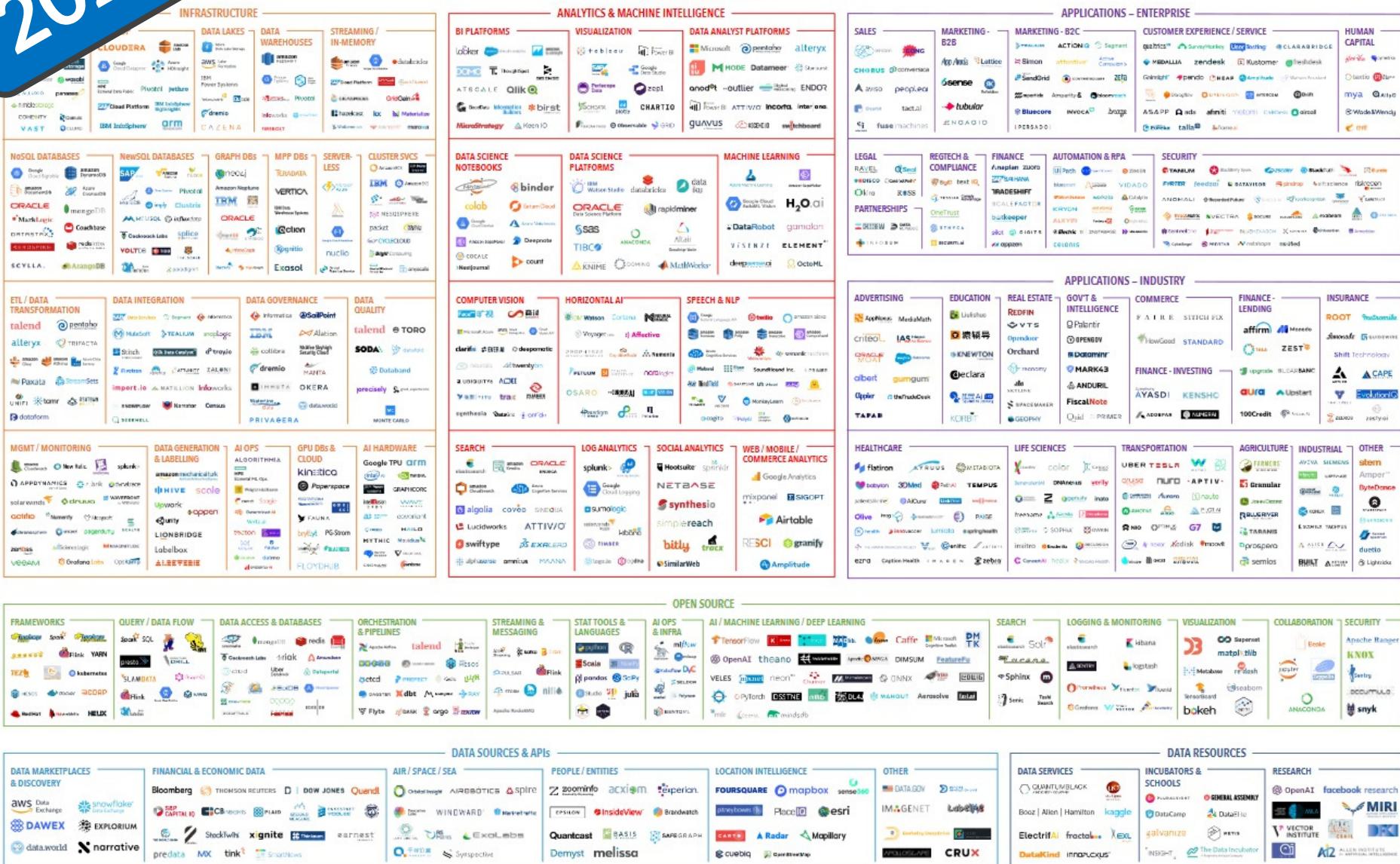
2018

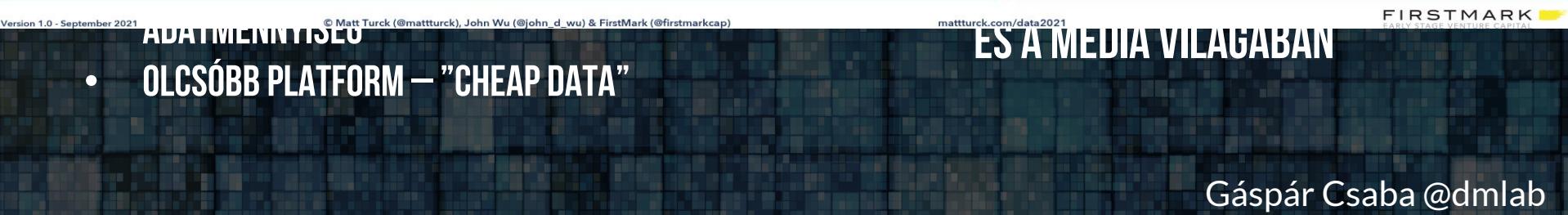
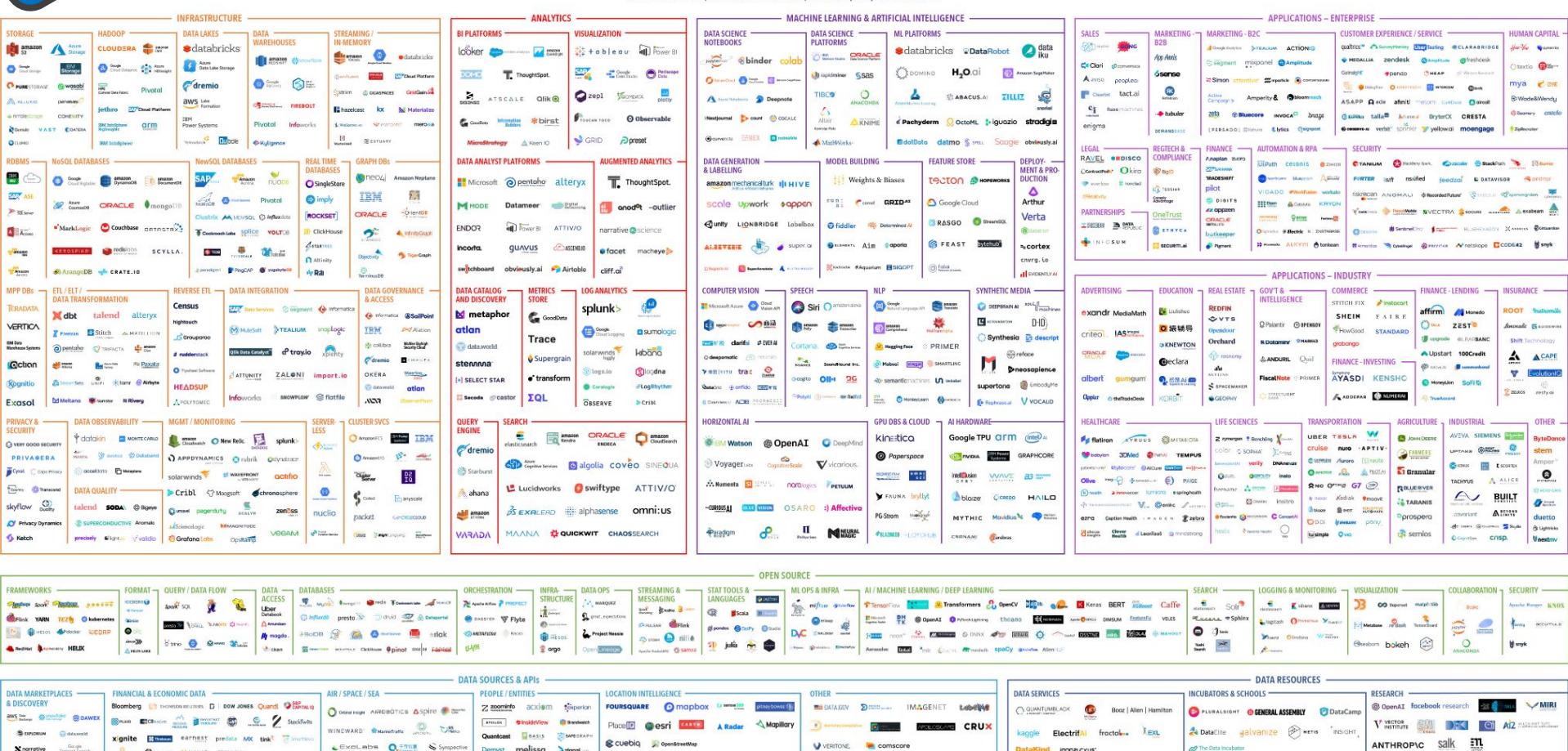
BIG DATA & AI LANDSCAPE 2018





2020

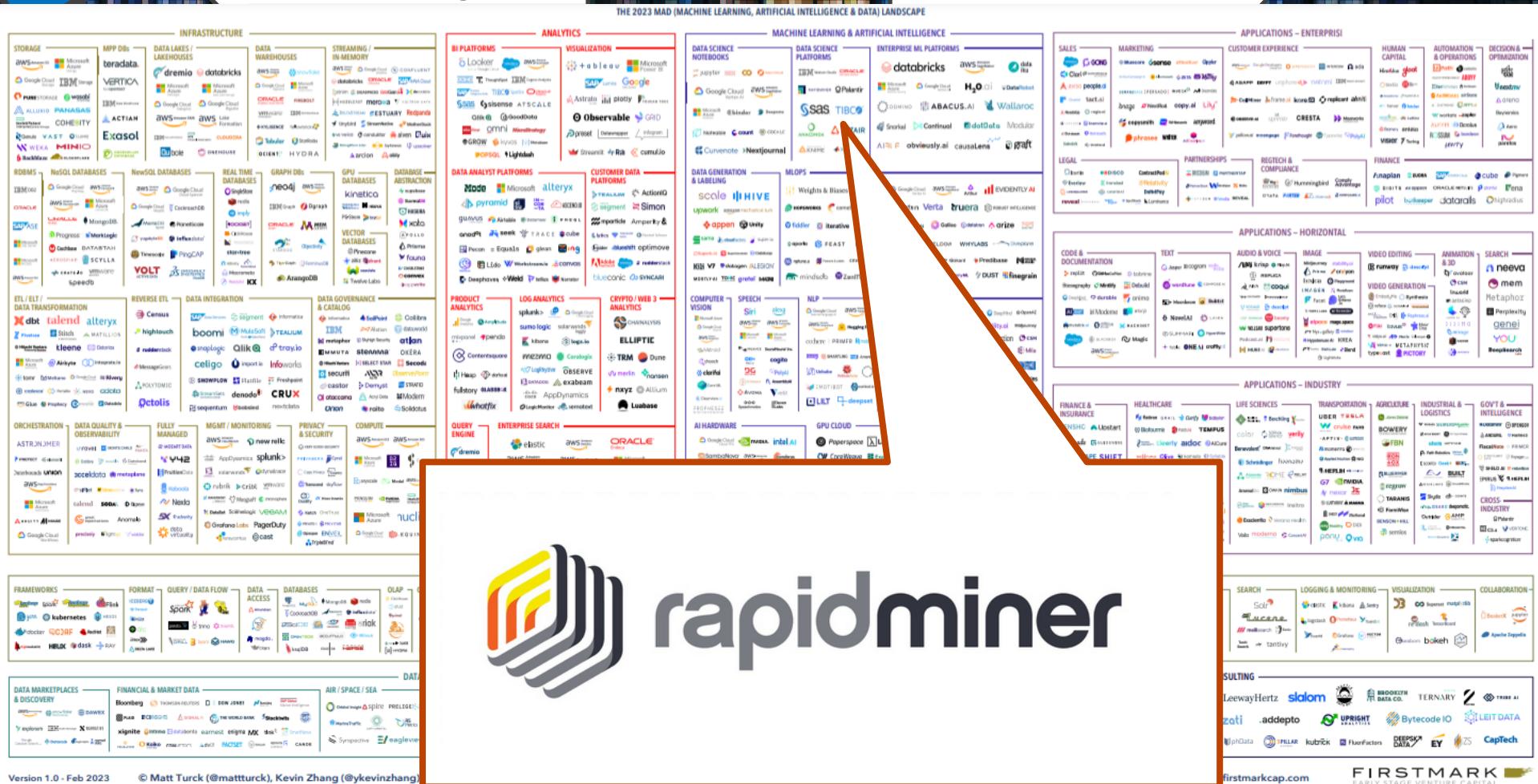




2023

Technológiai

BIG



Version 1.0 - Feb 2023

© Matt Turck (@mattturck), Kevin Zhang (@ykevinzhang)

- ADATUMINTRO
- OLCSÓBB PLATFORM – "CHEAP DATA"

ES A MEDIA VILÁGÁBAN

Gáspár Csaba @dmlab

FIRSTMARK
EARLY STAGE VENTURE CAPITAL



RADOOP STORY

Big Data analytics made easy



BIG DATA

Technológiai

ÚJ ADATTÁROLÁSI
ÉS -FELDOLGOZÁSI PARADIGMA



- NYÍLT FORRÁSKÓD, EXTRA NAGY ADATMENNYISÉG
- OLCSÓBB PLATFORM – "CHEAP DATA"

BIG DATA

Technológiai

Üzleti

ÚJ ADATTÁROLÁSI
ÉS -FELDOLGOZÁSI PARADIGMA

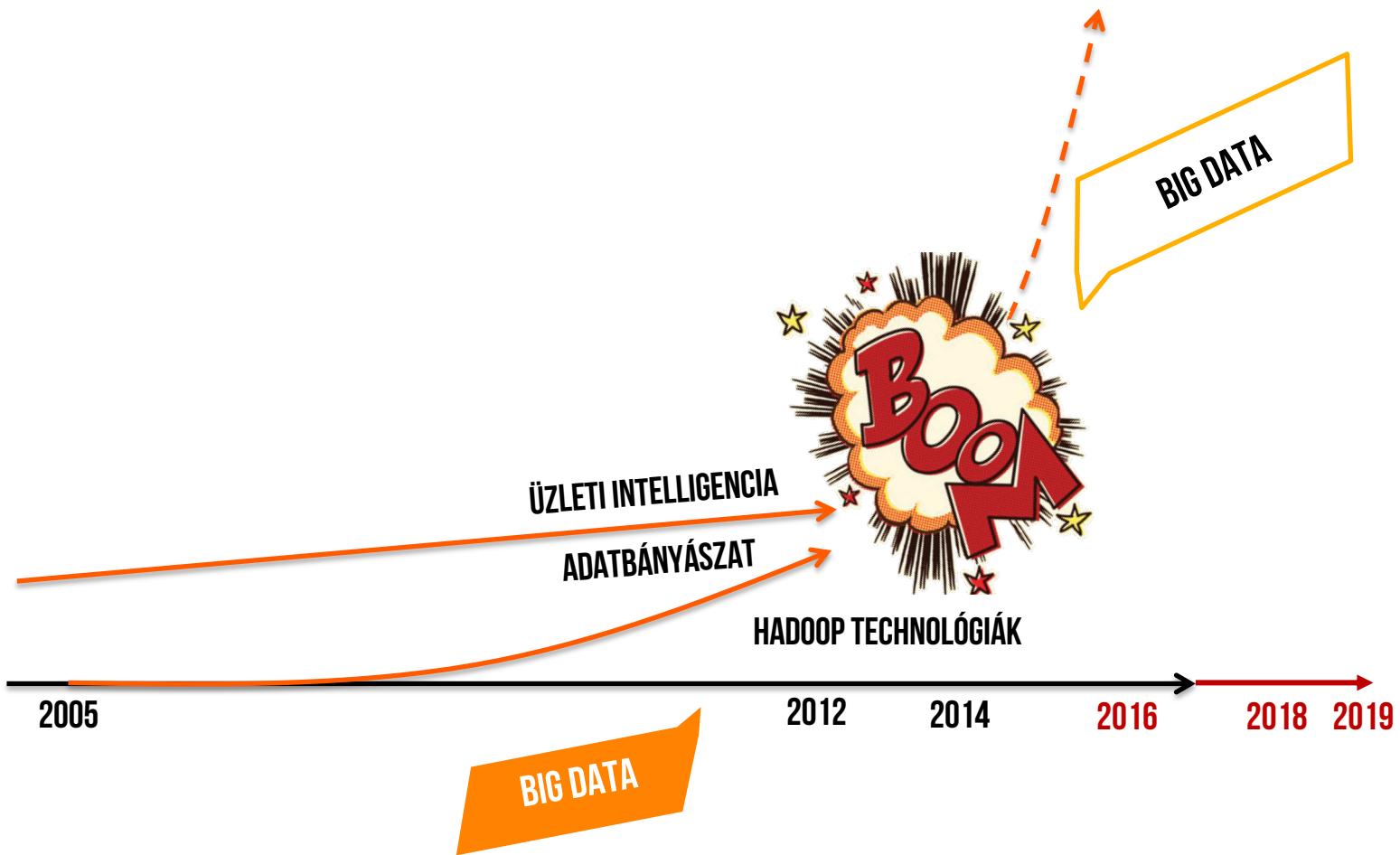


ADATOK ÚJSZERŰ,
INNOVATÍV FELHASZNÁLÁSA



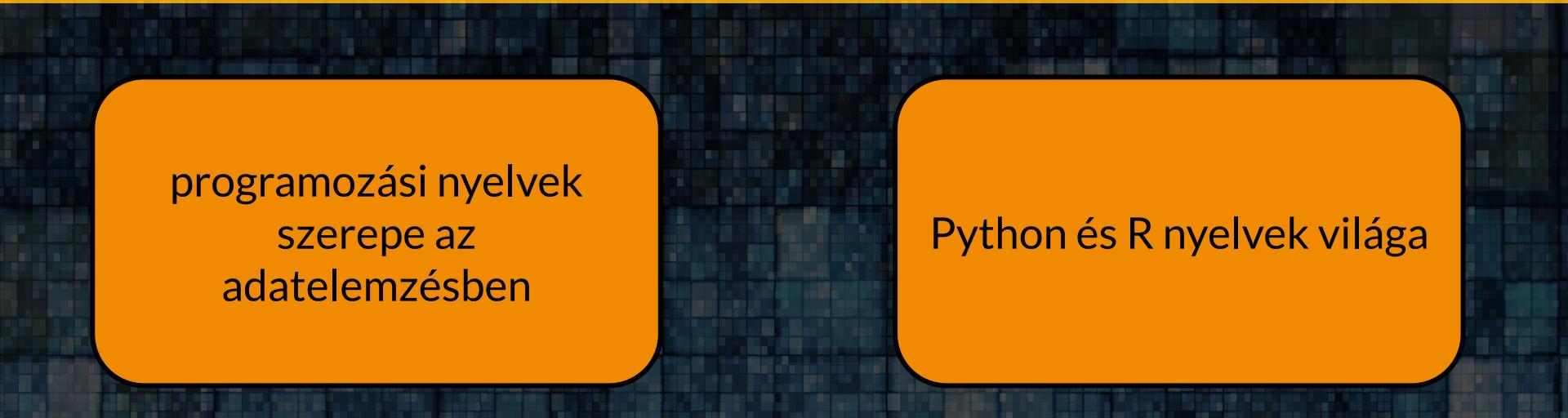
- NYÍLT FORRÁSKÓD, EXTRA NAGY ADATMENNYISÉG
- OLCSÓBB PLATFORM – "CHEAP DATA"

- ADATVAGYON KIHASZNÁLÁSA VAGY
- CÉLZOTT ADATGYŰJTÉS
- PREDIKTÍV ANALITIKA





"A világ állandóan változik,
és Darwin óta tudjuk,
hogy nem a legokosabb, nem a legerősebb, de nem is a legügyesebb
lesz az, aki ezekből a változásokból a legjobban jön ki.
Hanem az, aki legjobban alkalmazkodik a változáshoz."



programozási nyelvek
szerepe az
adatelemzésben

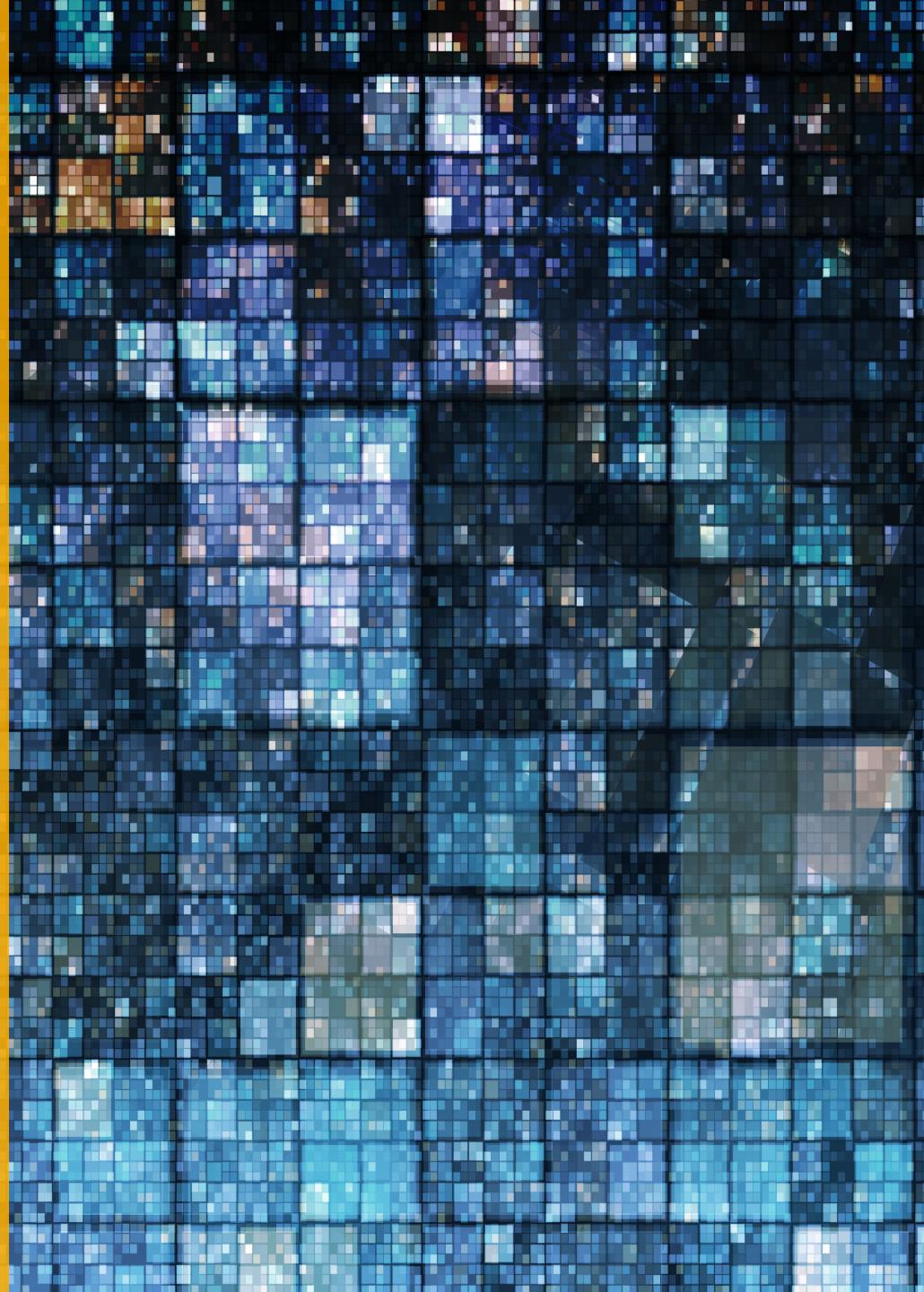
Python és R nyelvek világa

Relációs adatbázisok szemlélete

- Forrásadatok irányába
- „Adattábla” fogalom
 - Egy entitás – egy sor
 - Strukturált adatok

Előnyök / Hátányok

SQL és
adatbáziskezelők világa



Adatelemzési célnyelvek

- Saját adatreprezentációk
- Adatmanipulációra, modellezésre, majd vizualizációra specializált

Előnyök / Hátányok

- Kiemelve: zártság

SAS és SPSS nyelvek

SQL és
adatbáziskezelők világa

Adatfolyam jelleg

- Jól átlátható folyamatok
 - Kicsi utasításkészlet
- Vizualizáció kiemelt szerepe
- Üzleti felhasználók fele való nyitás

Előnyök / Hátányok

- Kiemelve:
Előrecsomagolt

Data flow alapú grafikus
felületek
(SAS Enterprise Miner, Guide,
SPSS Clementine / Modeler)

SAS és SPSS nyelvek

SQL és
adatbáziskezelők világa

Adatfolyam jelleg

- Kiterjesztett utasításkészlet
- Szabadon bővíthető
- Közösségek erejének megjelenése
- Modernebb algoritmusok

Előnyök / Hátányok

- Kiemelve kiforrasztan felület

Open source grafikus felületek
(Weka, RapidMiner, KNime)

Data flow alapú grafikus
felületek
(SAS Enterprise Miner, Guide,
SPSS Clementine / Modeler)

SAS és SPSS nyelvek

SQL és
adatbáziskezelők világa

Újra programozási irány

- Integrálhatóság, sokfajta adat kezelése
- Nagyobb szabadság a vizualizációban
- Funkcionális programozás
- Közösség erejére támaszkodik
- BIG DATA hullám

Pyhont és R nyelvek

Open source grafikus felületek
(Weka, RapidMiner, KNime)

Data flow alapú grafikus
felületek
(SAS Enterprise Miner, Guide,
SPSS Clementine / Modeler)

SAS és SPSS nyelvek

SQL és
adatbáziskezelők világa



PYTHON NYELVRŐL





PYTHON

Start

- 1991
- Guido van Rossum (holland programozó – 2005 Google – 2012 Dropbox)
- Hangsúly: kód olvashatósága + gyors fejlesztés (- adatelemzés)

Csomag központú

- Python Package index – PyPi
- Adatelemzős irány néhány éve
 - **Pandas** - adatszerkezetek, jó „adattábla” fogalom
 - **Numpy** - statisztikai csomag
 - **Scikit-learn** (sklearn) – gépi tanulás - **2013**



PYTHON

Start

- 1991
- Guido van Rossum (holland programozó – 2005 Google – 2012 Dropbox)
- Hangsúly: kód olvashatósága + gyors fejlesztés (- adatelemzés)

Csomag központú

- Python Package index – PyPi
- Adatelemzős irány néhány éve
 - **Pandas** - adatszerkezetek, jó „adattábla” fogalom
 - **Numpy** - statisztikai csomag
 - **Scikit-learn** (sklearn) – gépi tanulás

```
>>> from sklearn import datasets  
>>> from sklearn.svm import SVC  
>>> iris = datasets.load_iris()  
>>> clf = SVC()  
>>> clf.fit(iris.data, iris.target)  
>>> clf.predict(iris.data[:3])
```

Interpretált
nyelv
(Prompt)



PYTHON

Környezet

- minden platformra könnyen installálható
- Anaconda – python alapú analitikai platform
 - Ingyenes – (supporttal: Anaconda Pro – 10.000\$ - 10 user / év)

Fejlesztői környezetek

- Spyder
- iPython Notebook



SPYDER

Spyder

Editor - /Users/rob/Desktop/qutip-official/examples/ex_floquet_quasienergies.py

```
48 #quasi_energies = zeros([len(A_vec), 2])
49 #f_gnd_prob = zeros([len(A_vec), 2])
50 quasi_energies = zeros([len(eps0_vec), 2])
51 f_gnd_prob = zeros([len(eps0_vec), 2])
52 wf_gnd_prob = zeros([len(eps0_vec), 2])
53
54 for idx, eps0 in enumerate(eps0_vec):
55
56     H0 = - delta/2.0 * sx - eps0/2.0 * sz
57     H1 = A/2.0 * sz
58
59     # H = H0 + H1 * sin(w * t) in the 'list-string' format
60     H = [H0, [H1, 'sin(w * t)']]
61     Hargs = {'w': omega}
62
63     # find the floquet modes
64     f_modes, f_energies = floquet_modes(H, T, Hargs)
65
66     print "Floquet quasienergies[%d, :]" % idx, f_energies
67
68     quasi_energies[idx, :] = f_energies
69
70     f_gnd_prob[idx, 0] = expect(sm.dag() * sm, f_modes[0])
71     f_gnd_prob[idx, 1] = expect(sm.dag() * sm, f_modes[1])
72
73     f_states = floquet_states_t(f_modes, f_energies, 0, H, T, Hargs)
74
75     wf_gnd_prob[idx, 0] = expect(sm.dag() * sm, f_states[0])
76     wf_gnd_prob[idx, 1] = expect(sm.dag() * sm, f_states[1])
77
78 return quasi_energies, f_gnd_prob, wf_gnd_prob
79
80 #
81 # set up the calculation: a strongly driven two-level system
82 # (repeated LZ transitions)
83 #
84 delta = 0.2 * 2 * pi # qubit sigma_x coefficient
85 eps0 = 0.5 * 2 * pi # qubit sigma_z coefficient
86 gamma1 = 0.0 # relaxation rate
87 gamma2 = 0.0 # dephasing rate
88 A = 2.0 * 2 * pi
```

Variable explorer

Name	Type	Size	Value
A	float	1	12.566370614359172
T	float	1	1.0
delta	float	1	1.2566370614359172
e	float	1	2.718281828459045
eps0	float	1	3.141592653589793
gamma1	float	1	0.0
gamma2	float	1	0.0
numpy_requirement	str	1	1.6.0
omega	float	1	6.283185307179586
pi	float	1	3.141592653589793
qutip_rc_file	str	1	/Users/rob/.qutiprc

Object inspector Variable explorer File explorer

Console

```
>>> eps0 = 0.5 * 2 * pi # qubit sigma_z coefficient
>>> gamma1 = 0.0 # relaxation rate
>>> gamma2 = 0.0 # dephasing rate
>>> A = 2.0 * 2 * pi
>>> psi0 = basis(2,0) # initial state
>>> omega = 1.0 * 2 * pi # driving frequency
>>> T = (2*pi)/omega # driving period
>>>
>>> H0 = - delta/2.0 * sx - eps0/2.0 * sz
>>> H0
Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True
Qobj data =
[[ -1.57079633 -0.62831853]
 [-0.62831853  1.57079633]]
>>> |
```

Permissions: RW End-of-lines: LF Encoding: UTF-8-GUESSED Line: 58 Column: 8

IPYTHON NOTEBOOK

Környezet

- Minden IP[y]: Notebook
- Anaconda
 - Inguru

Fejlesztői

- Spyder
- iPython

user / év)

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kn} \quad k = 0, \dots, N - 1$$

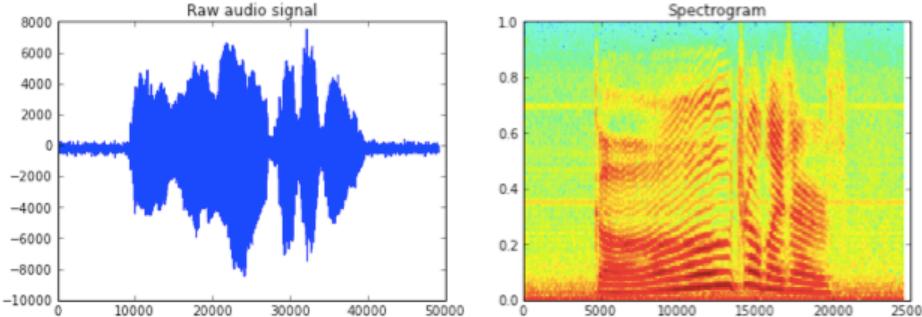
using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin specgram routine:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```





PYTHON

Környezet

- minden platformra könnyen installálható
- Anaconda – python alapú analitikai platform
 - Ingyenes – (supporttal: Anaconda Pro – 10.000\$ - 10 user / év)

Fejlesztői környezetek

- Spyder
- iPython Notebook

- Névválasztás: Monty Python repülő církusza
- Első programozási nyelvként népszerű
- Több big data technológia programozható vele (pl. Spark)



R NYELVRŐL



R NYELV

Start

- 1996
- Ross Ihara és Robert Gentleman (új-zélandi és kanadai statisztikusok – akadémiai környzet)
- Hangsúly: statisztikai számítások és vizualizáció – S nyelv alapján

Csomag központú

- CRAN – egységes repository a meghatározó csomagoknak
- Erős közösség (levelezési listák – közös dokumentumok – Stack Overflow csoportok)

Interpretált
nyelv
(Prompt)

```
dataDirectory <- "D:/"
data <- read.csv(paste(dataDirectory, 'regression.csv', sep=""), header = TRUE)
model <- lm(Y ~ X , data)
predictedY <- predict(model, data)
```



R NYELV

Környezet

- minden platformra könnyen installálható

Fejlesztői környezetek

- RStudio



RSTUDIO

Körny

- Mi
- Fejles
- RS

~/editor - RStudio

Report.Rnw x analysis.R* x rawdata x prep.R x clean x > Workspace History

Source on Save | Run | +

```
1 # User Analysis
2
3 setwd("~/analysis")
4 source("prep.R")
5
6 library(plyr)
7 library(lattice)
8 library(ggplot2)
9
10 # Import data set
11 rawdata <- read.csv("stats.csv")
12 totalUsers <- dim(rawdata)[1]
13
14 # Clean data set
15 clean <- prepareStats(rawdata)
16
17 # Subset of active users
18 active <- subset(clean, active == 1)
19 count(active,"daysSinceAccountCreated < 30")[2,2]
20 mean(active$age)
21
22
23
```

(Top Level) R Script

Console ~/analysis/

```
[1] 0.547505
> dim(active)
[1] 197323     35
> mean(clean$age)
[1] 33.89018
> summary(active$age)
   Min. 1st Qu. Median 3rd Qu. Max.
14.00  27.00  34.00  35.76  43.00  87.00
> count(active, "age > 25")
age...25 freq
1 FALSE 38045
2 TRUE 159278
>
```

Workspace

- portfolio.R
- portfolioStats.R
- Q1Report.Rnw
- analysis.R
- rawdata
- prep.R
- clean

History

30Day	0.25
	47886L
	FALSE
	4
	Date[1]
	0.23
	integer[2]
validUsers	34
	360404L

Functions

```
prepareStats(data, sampleSize = NA)
```

Files Plots Packages Help

New Folder Delete Rename More

Home > analysis

Name	Size	Modified
..		
analysis.R	2.8 KB	Jan 6, 2011, 9:43 AM
prep.R	1.6 KB	Jan 7, 2011, 10:41 AM
stats.csv	86.5 MB	Jan 3, 2011, 11:22 AM



R NYELV

Környezet

- minden platformra könnyen installálható

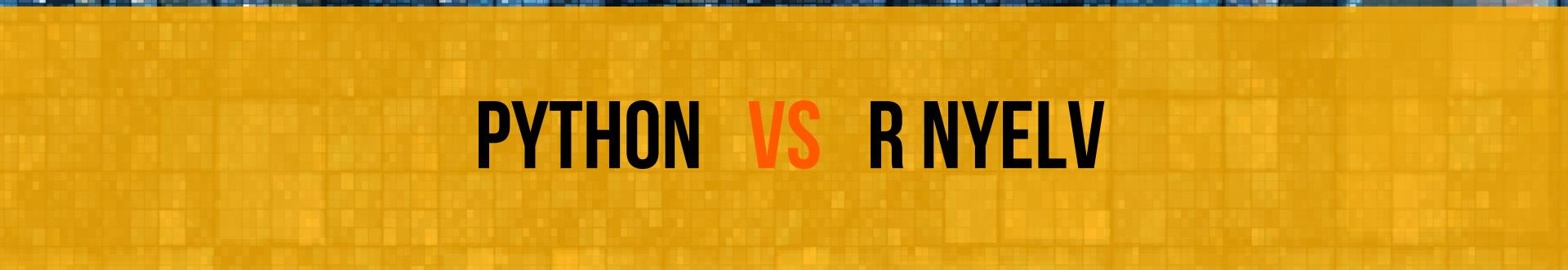
Fejlesztői környezetek

- RStudio

- Névválasztás: S nyelv után az R nyelv jön – nehéz keresni - #rstats
- „The worst thing about R is that... it was developed by statisticians.”



PYTHON VS R NYELV





VS.

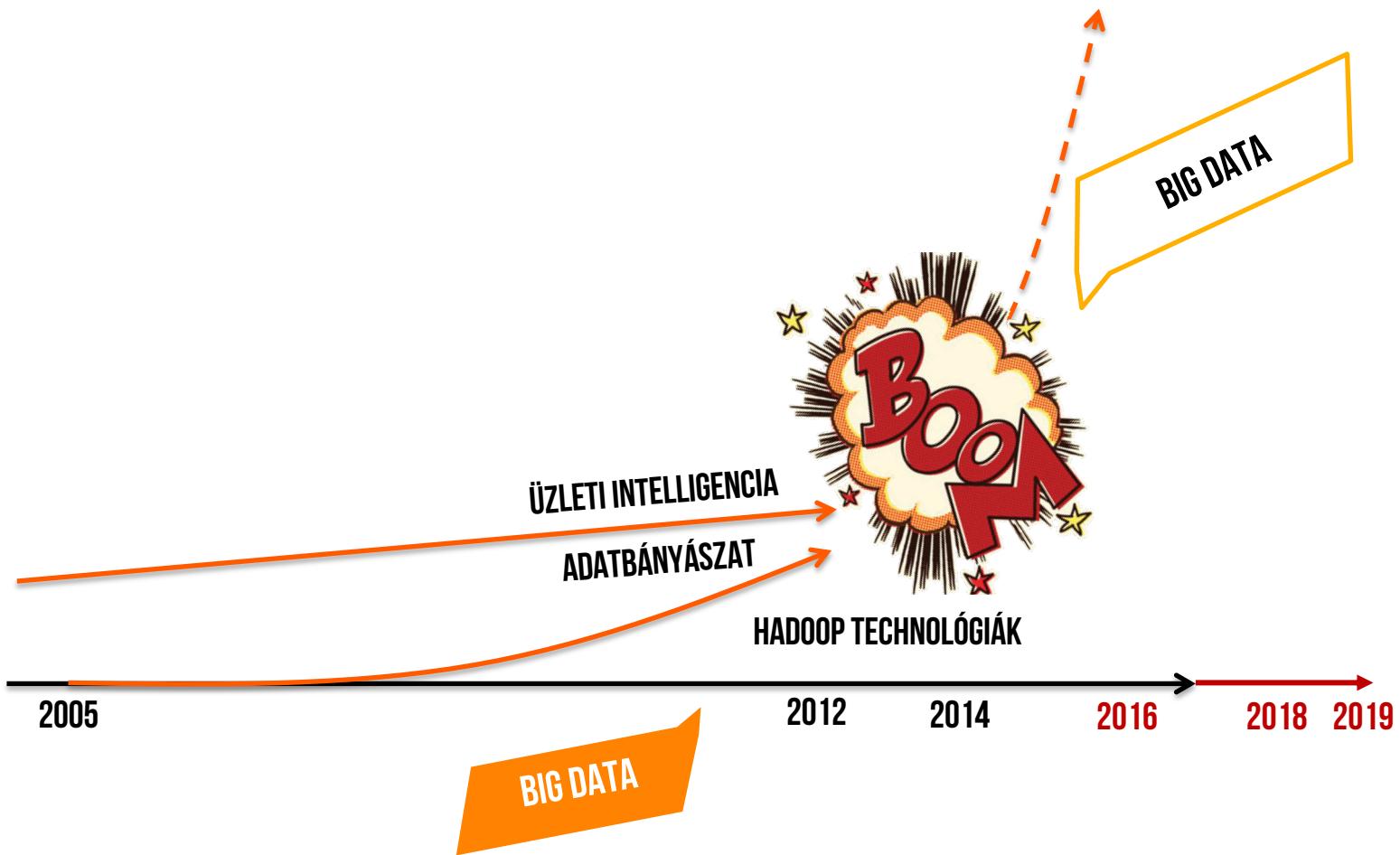


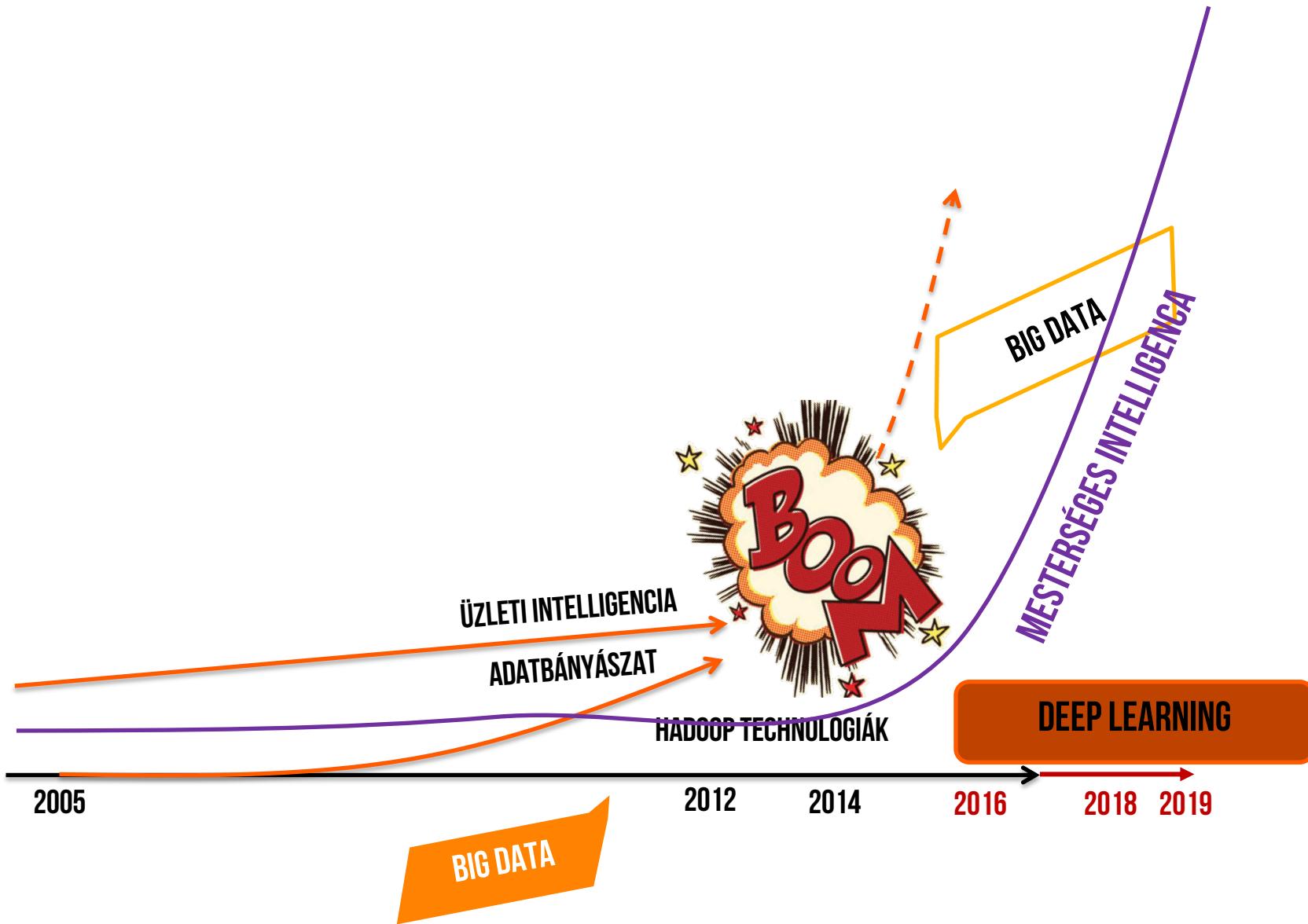
python

R and Python are waging war:
while both programming languages are gaining prominence
in the data analytics community, they are fighting
to become data scientists' language of choice.

Which side are you taking?

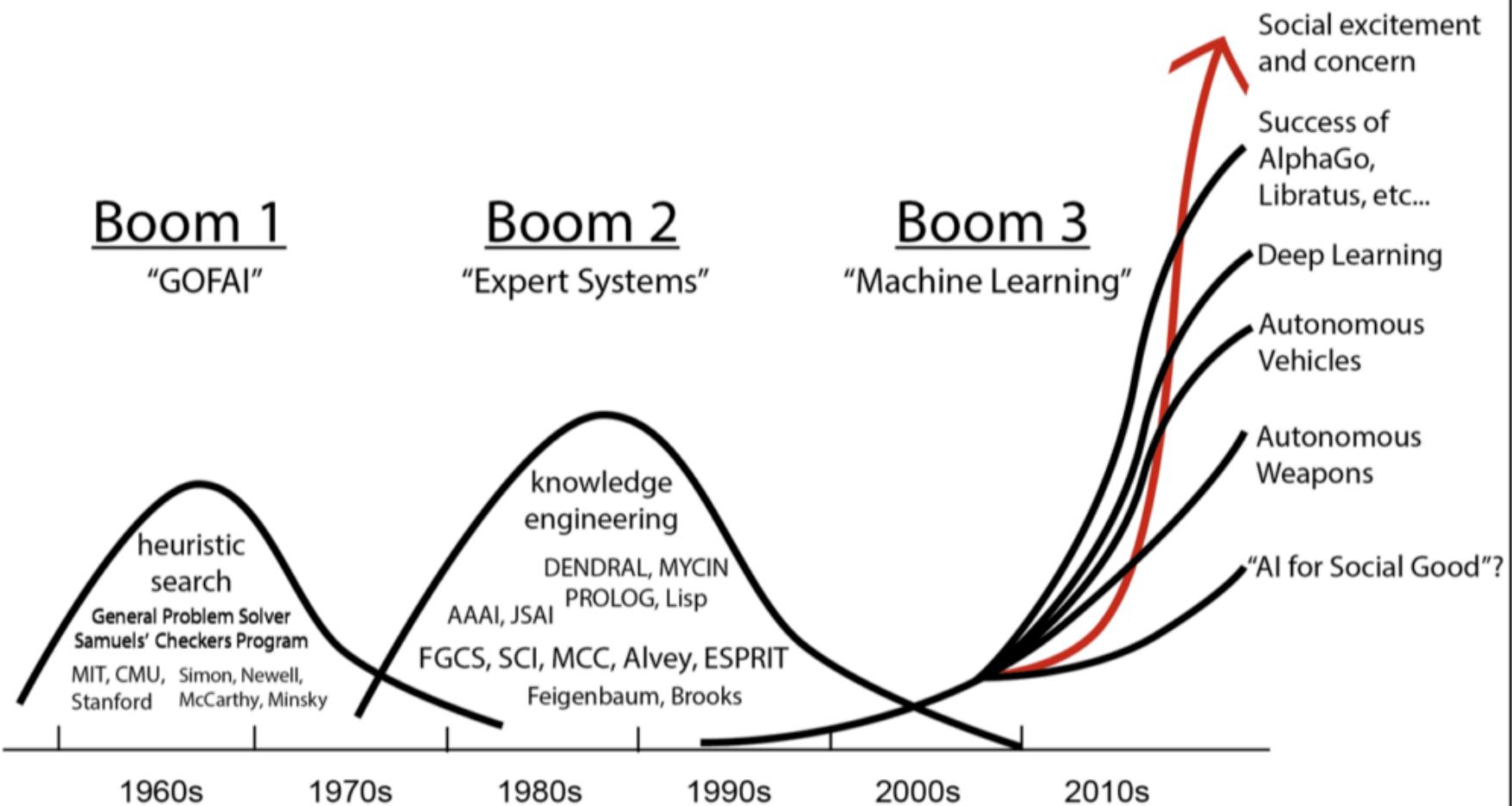






MESTERSÉGES INTELLIGENCIA

ARTIFICAL INTELLIGENCE



MESTERSÉGES INTELLIGENCIA HATÁSA A SZERVEZETEKRE



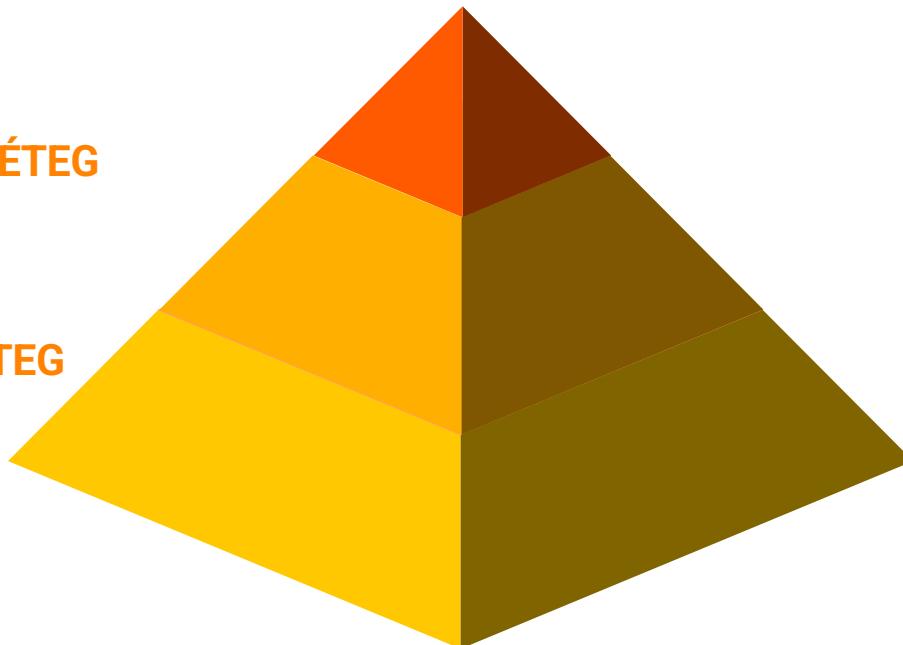
FELSŐ VEZETÉS



KÖZÉPVEZETŐI RÉTEG



VÉGREHAJTÓ RÉTEG



MESTERSÉGES INTELLIGENCIA HATÁSA A SZERVEZETEKRE



FELSŐ VEZETÉS

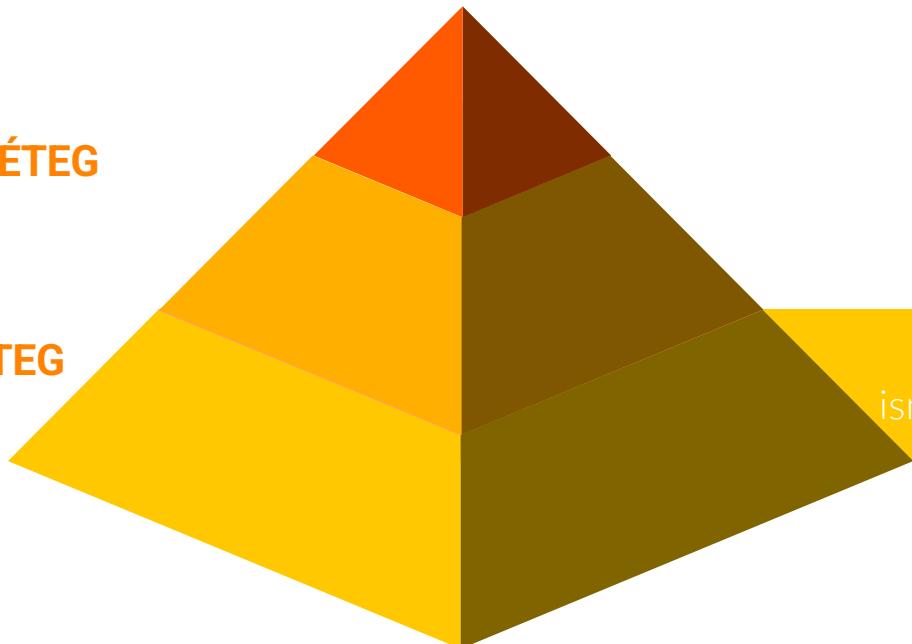


KÖZÉPVEZETŐI RÉTEG



VÉGREHAJTÓ RÉTEG

Automatizáció



Mely tevékenységek
ismétlődnek, helyettesíthetők?

MESTERSÉGES INTELLIGENCIA HATÁSA A SZERVEZETEKRE



FELSŐ VEZETÉS



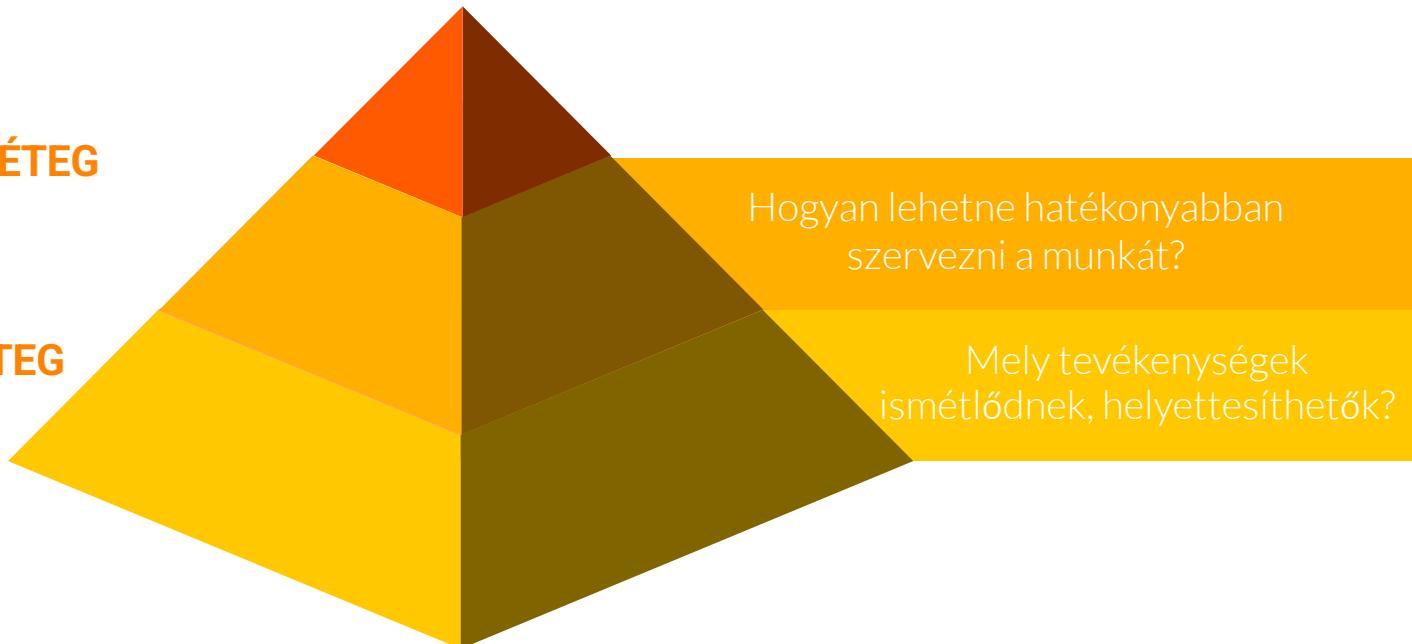
KÖZÉPVEZETŐI RÉTEG

Fejlett analitika



VÉGREHAJTÓ RÉTEG

Automatizáció



MESTERSÉGES INTELLIGENCIA HATÁSA A SZERVEZETEKRE



FELSŐ VEZETÉS

Adatvezérlelt döntéstámogatás



KÖZÉPVEZETŐI RÉTEG

Fejlett analitika



VÉGREHAJTÓ RÉTEG

Automatizáció



MÉGIS EGY KIS TÖRTÉNELEM

BIG FIVE

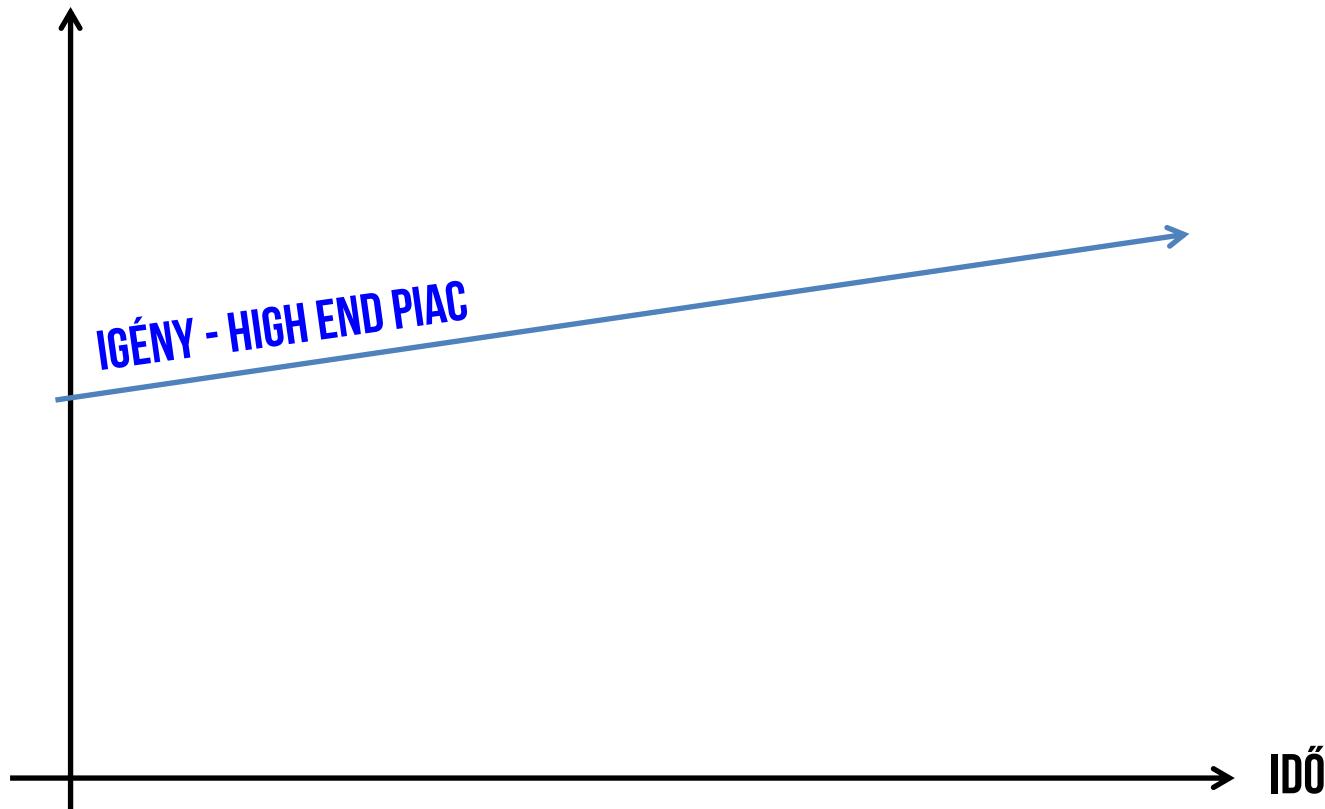
ROMBOLÓ INNOVÁCIÓ

Company	2010 Market Share (%)
SAP	23.0
Oracle	15.7
SAS Institute	13.2
IBM	11.6
Microsoft	8.7
Other Vendors	27.9



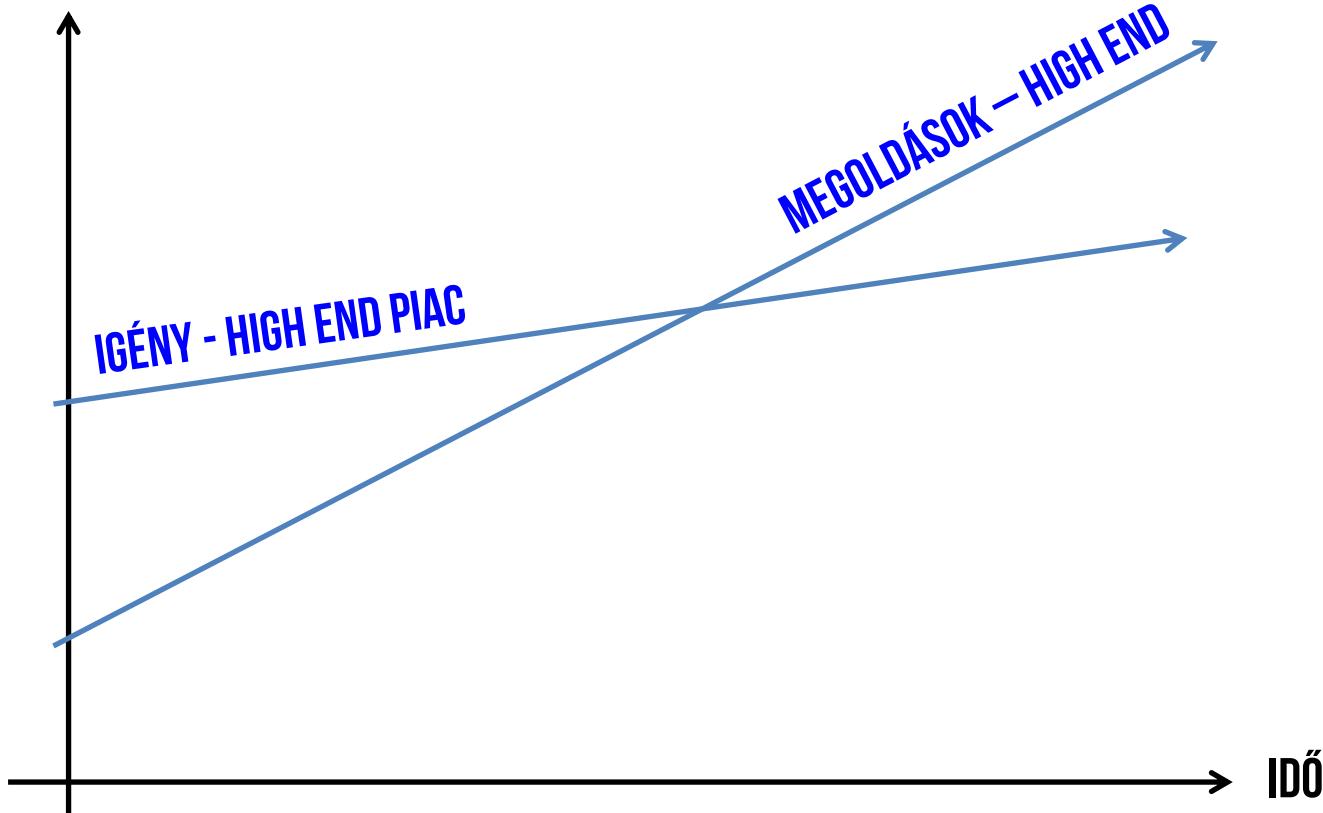
ROMBOLÓ INNOVÁCIÓ MŰKÖDÉSE

PERFORMANCE



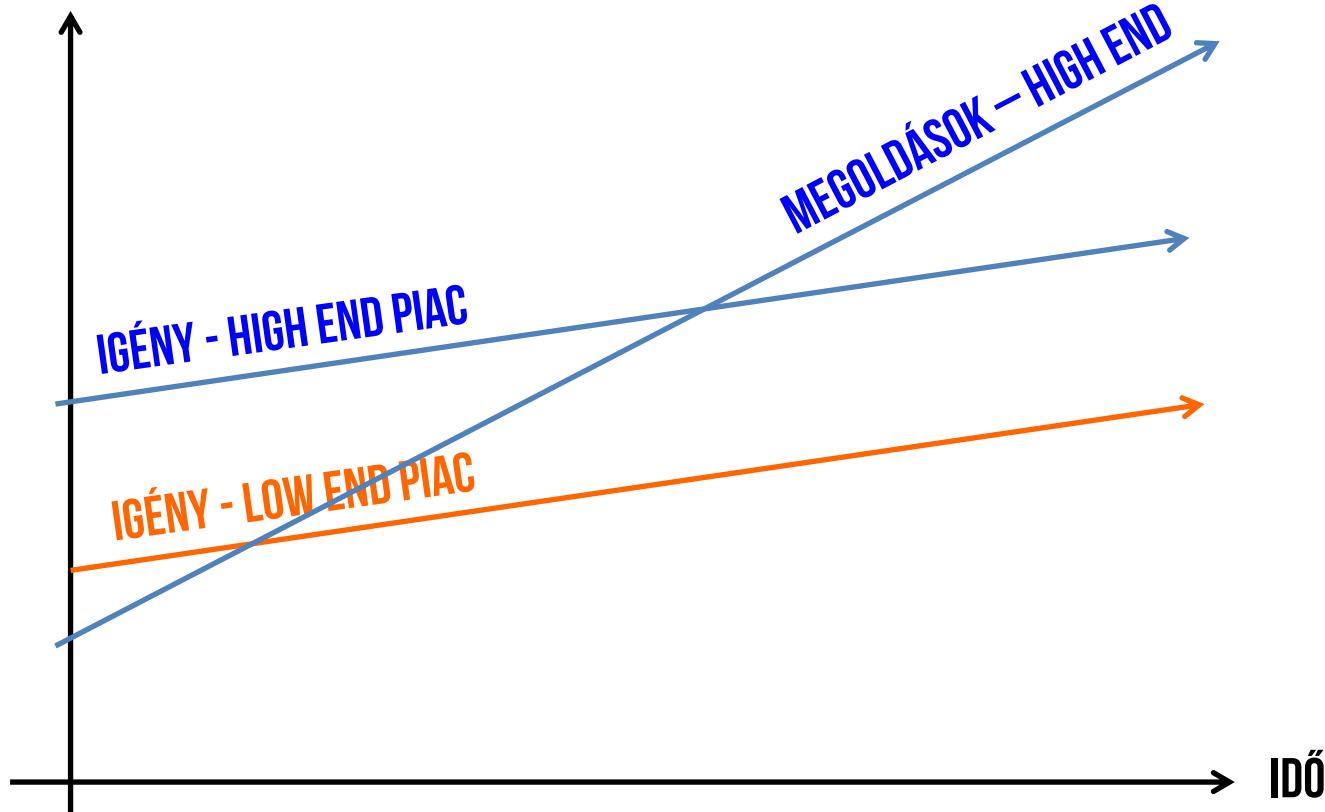
ROMBOLÓ INNOVÁCIÓ MŰKÖDÉSE

PERFORMANCE



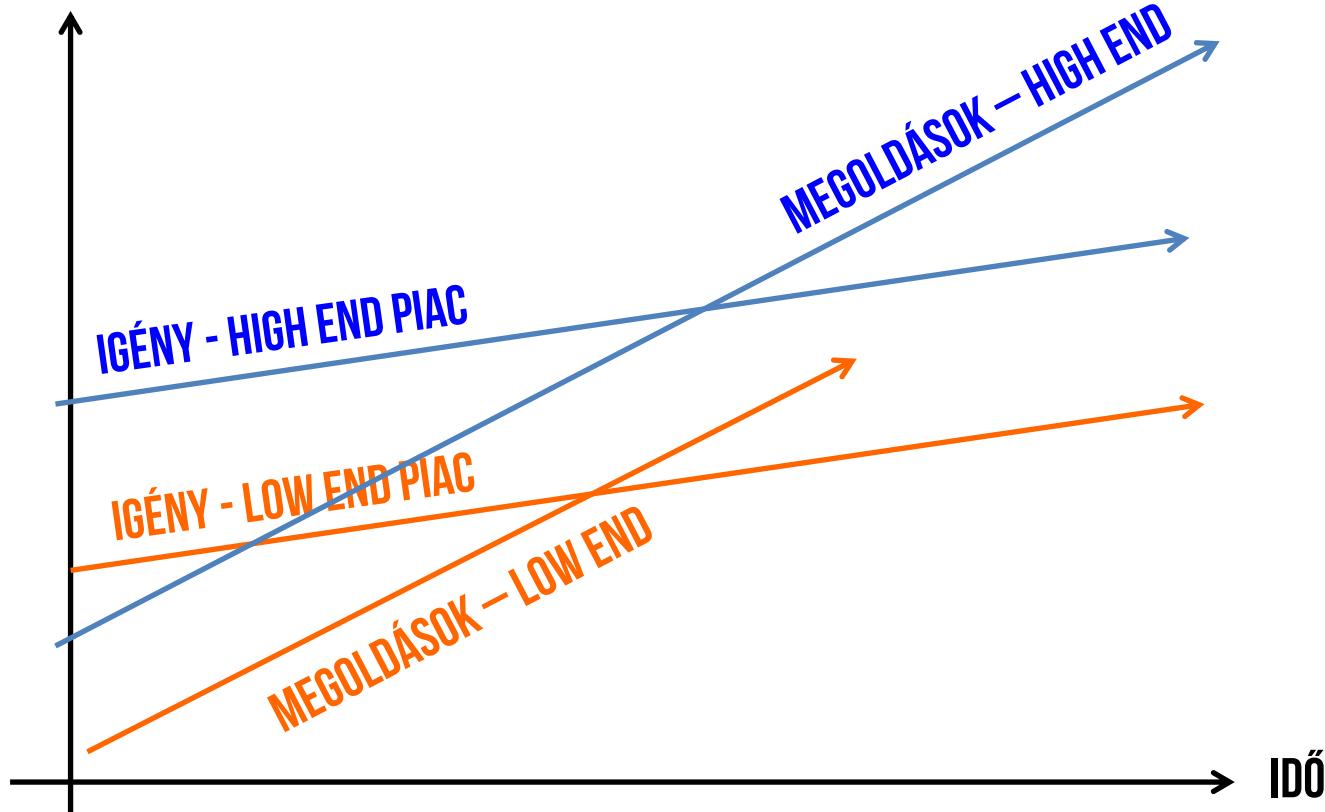
ROMBOLÓ INNOVÁCIÓ MŰKÖDÉSE

PERFORMANCE



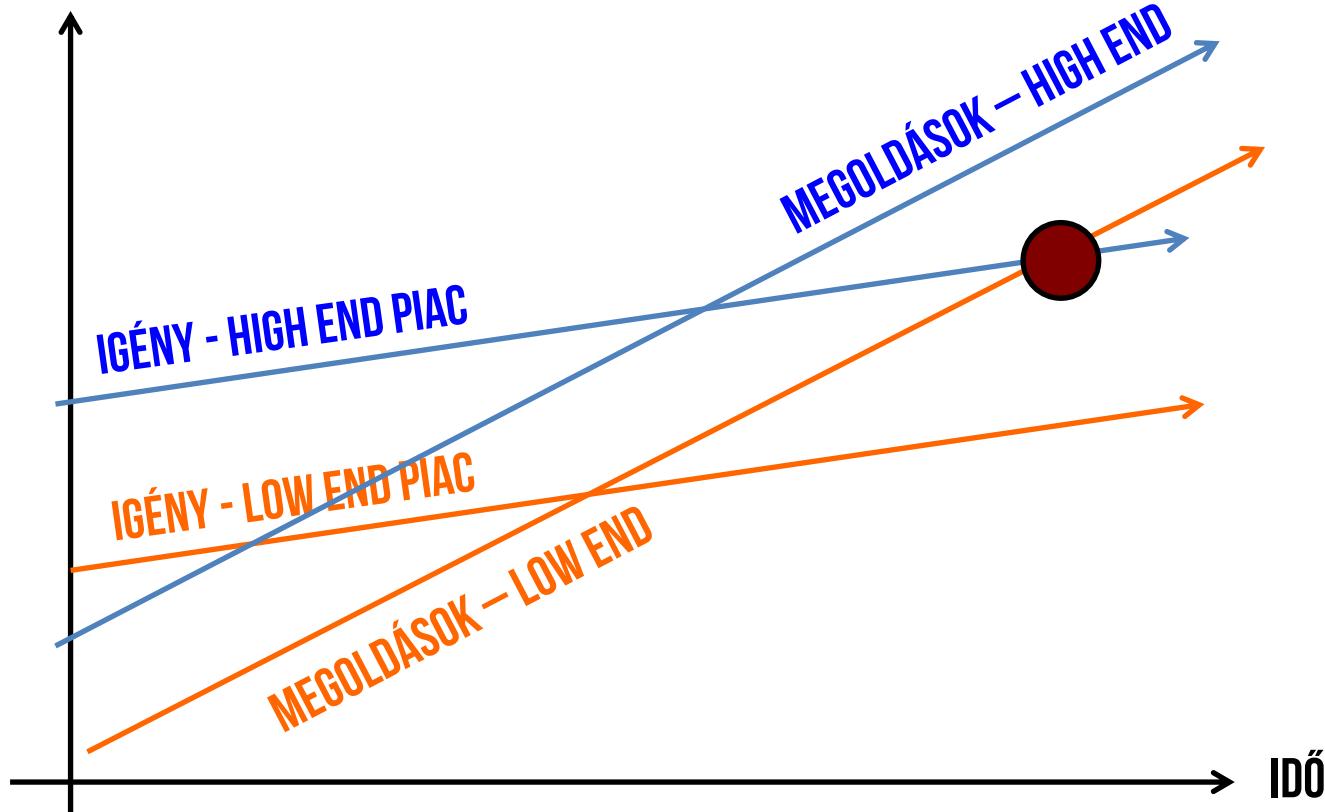
ROMBOLÓ INNOVÁCIÓ MŰKÖDÉSE

PERFORMANCE



ROMBOLÓ INNOVÁCIÓ MŰKÖDÉSE

PERFORMANCE



TARGET.COM STORY





A BIG DATA ÍGÉRETE



EHHEZ
KEPEST

...

A KULCS: DATA SCIENTIST



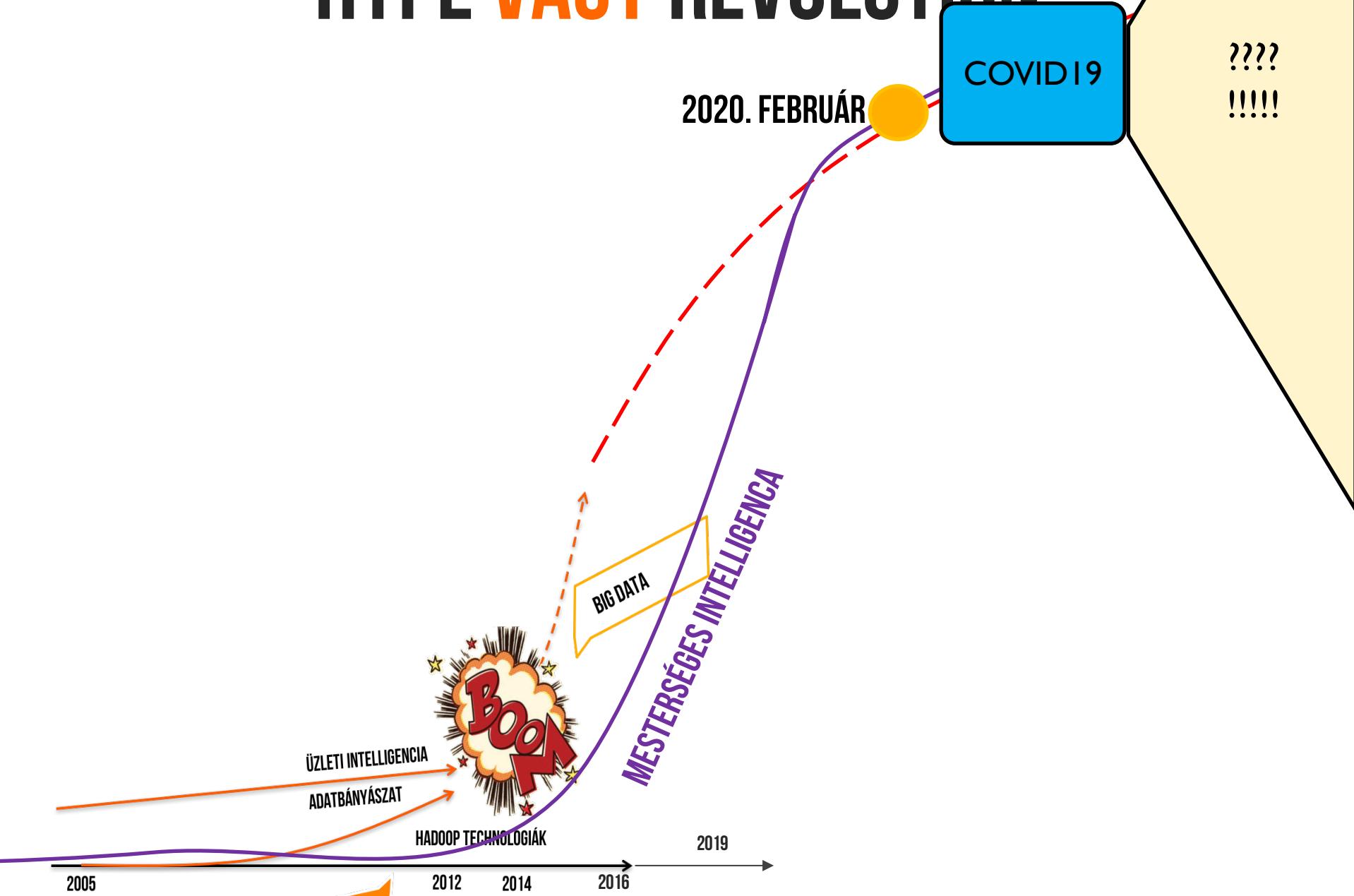
**MIHEZ
ÉRT A JÓ
DATA SCIENTIST?**



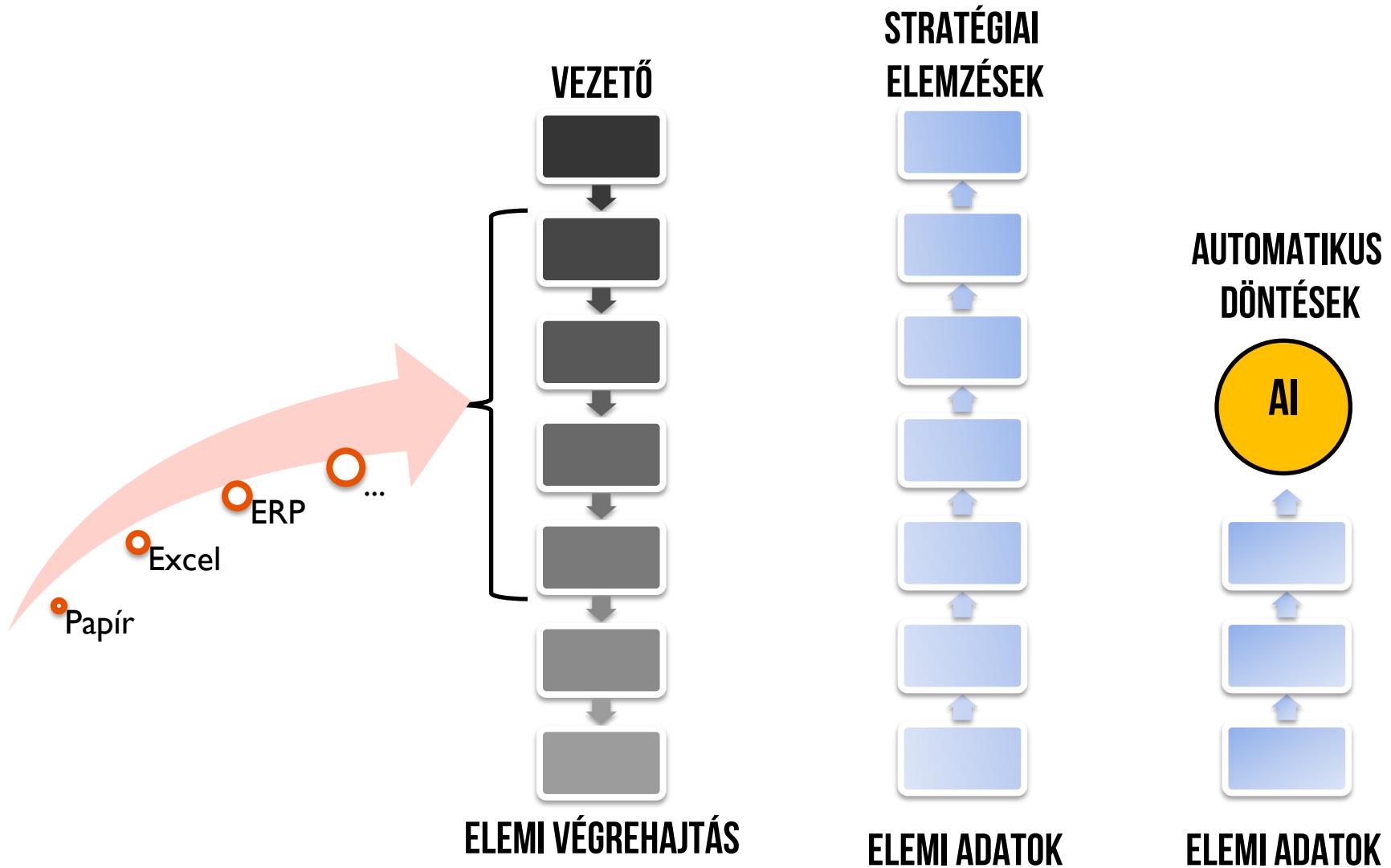


ÉS A BIG DATA BUMM UTÁN?

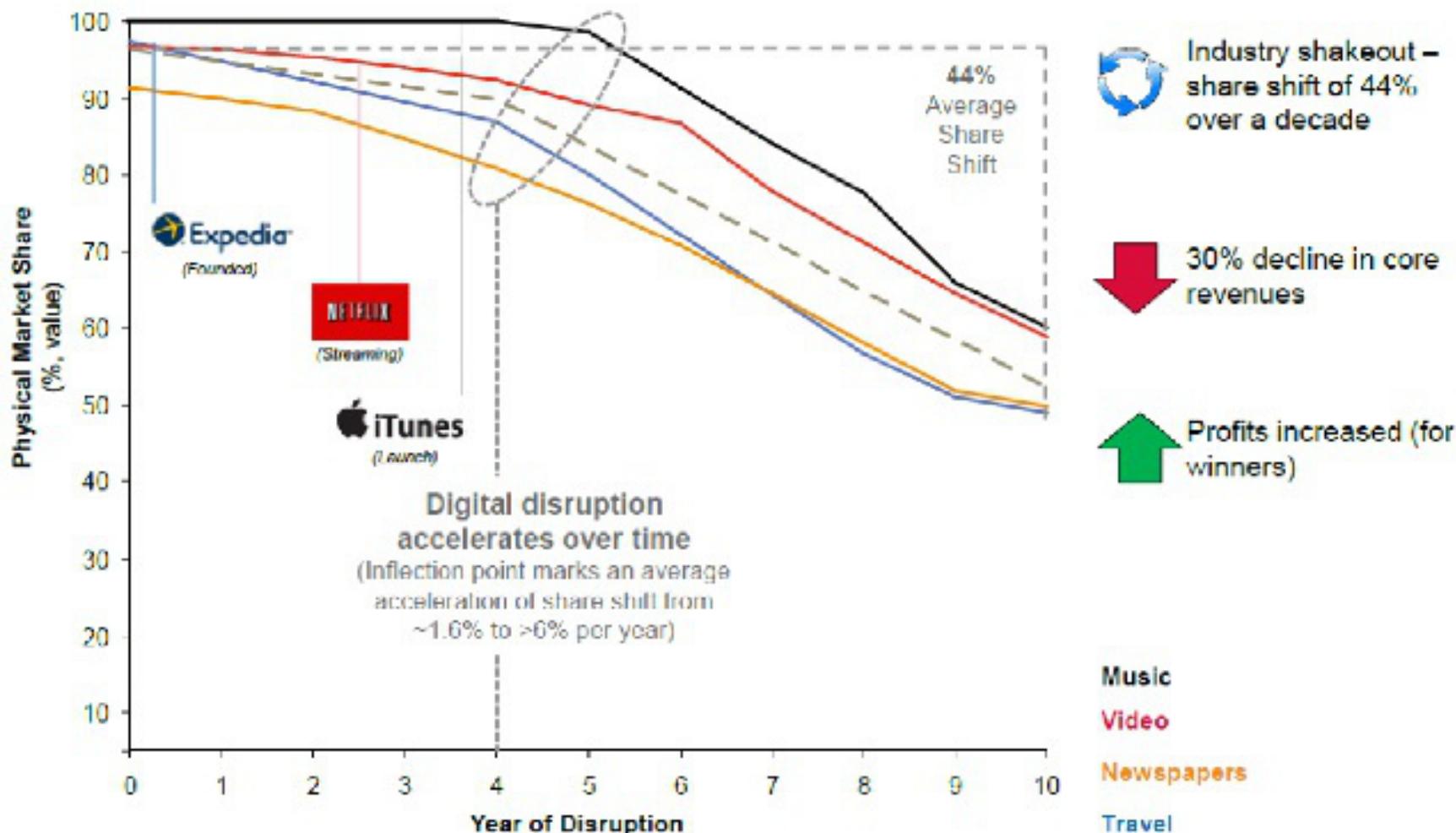
HYPE VAGY REVOLUTION



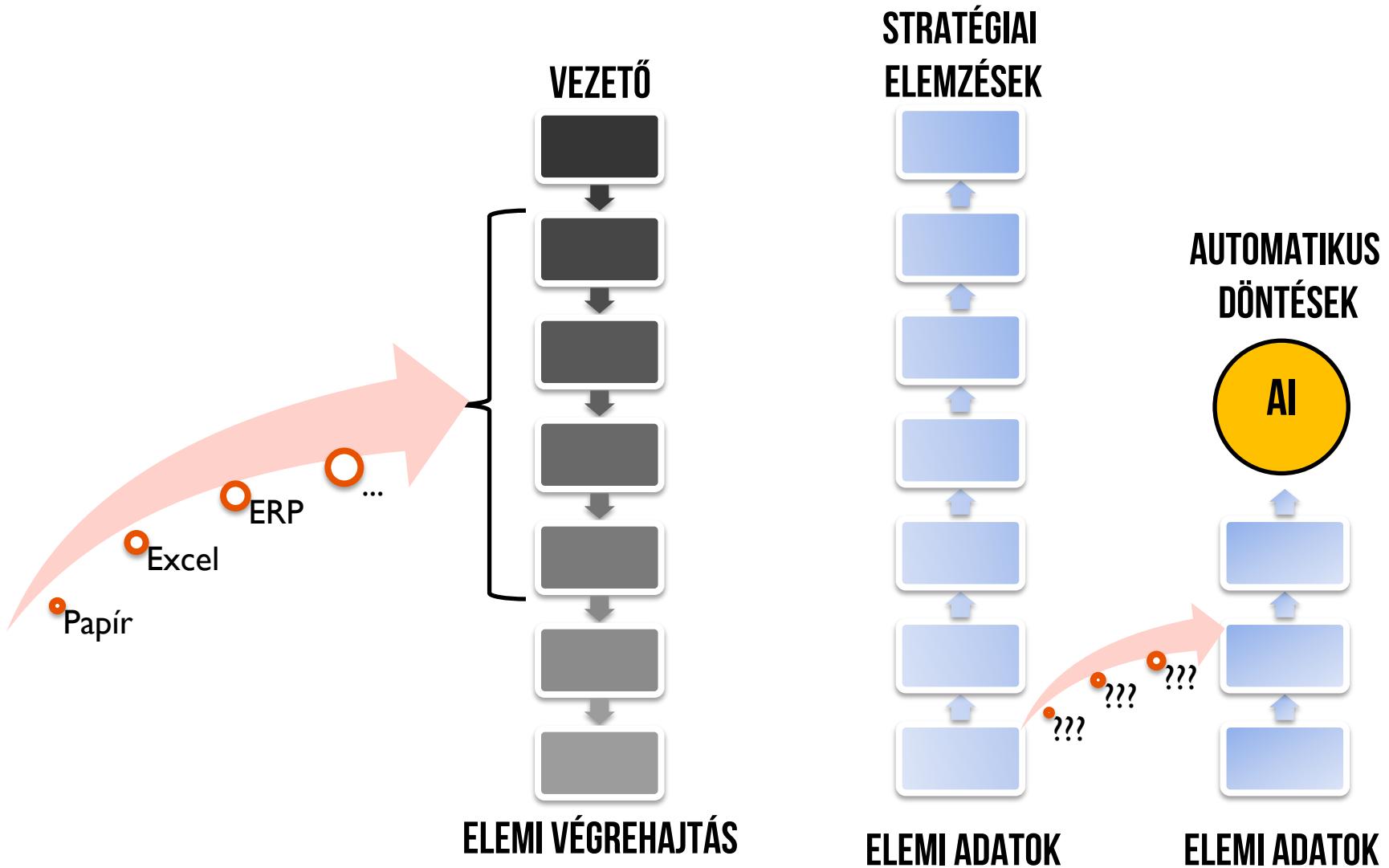
KI VEZÉRLI A RENDSZERT?



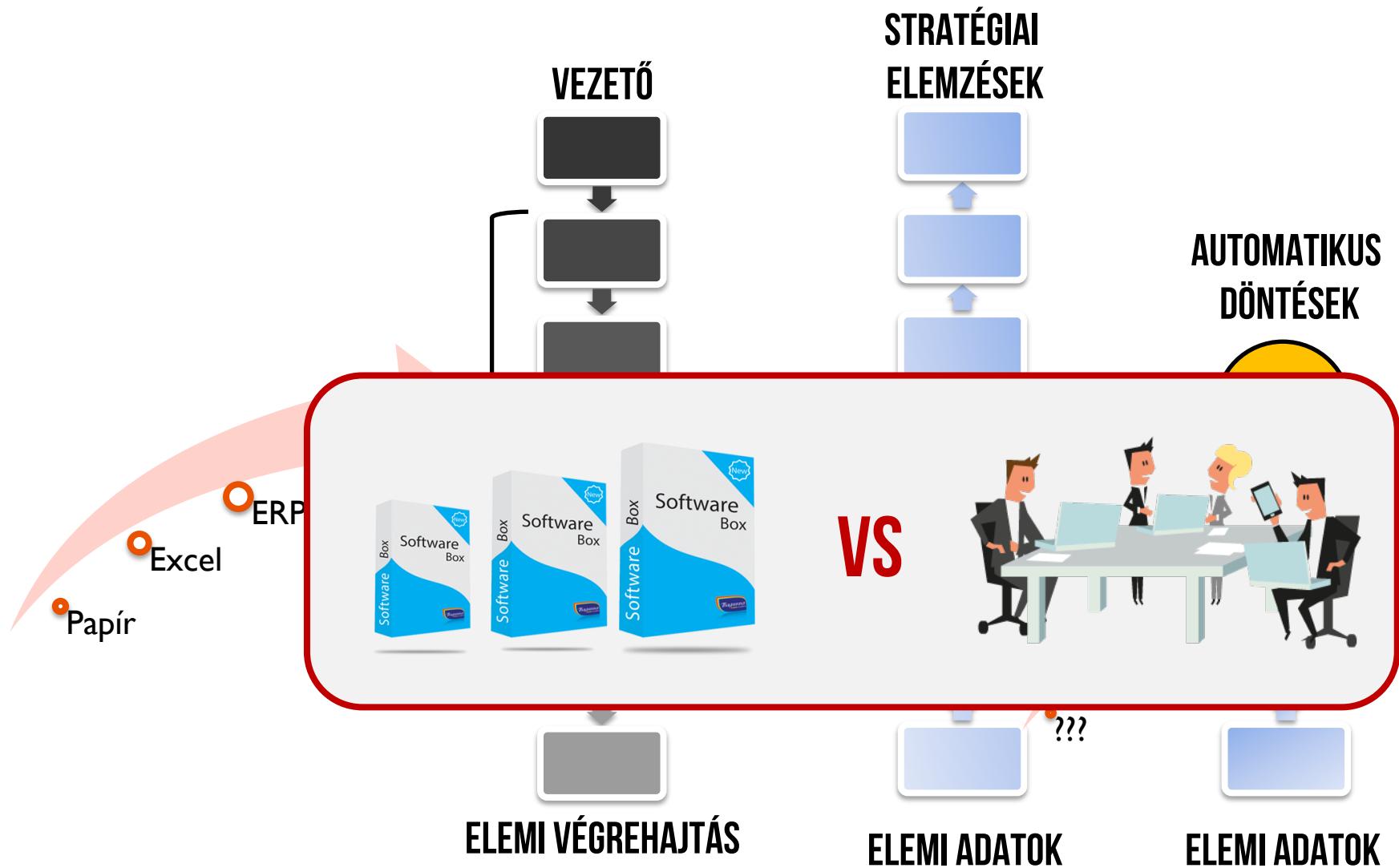
DIGITALIZÁCIÓ HATÁSAI



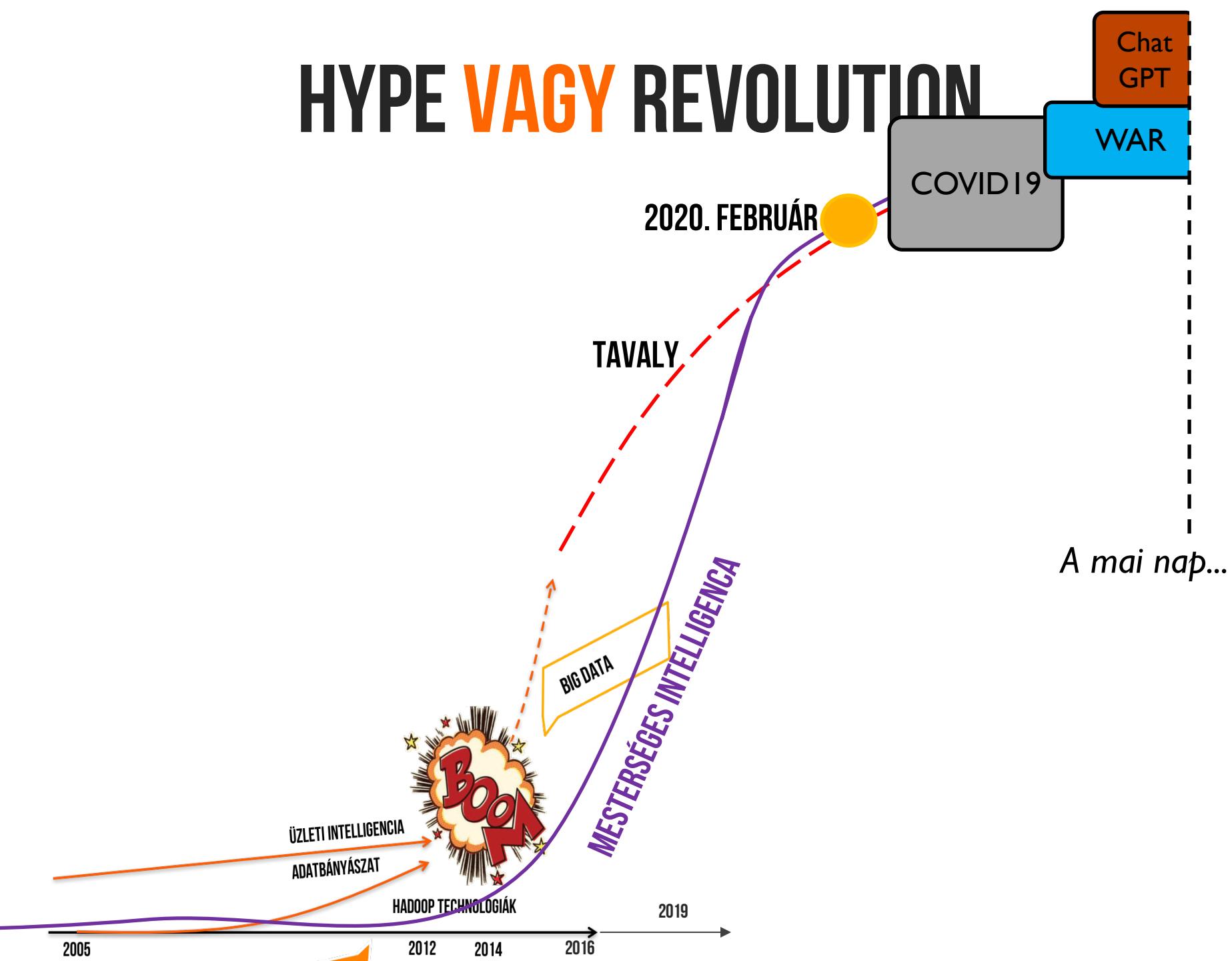
KI VEZÉRLI A RENDSZERT?



KI VEZÉRLI A RENDSZERT?



HYPE VAGY REVOLUTION





KÖVETKEZMÉNYEK – 3 IRÁNYBÓL

data oil

is the new

we need to find it,
extract it, refine it,
distribute it and
monetize it.

David Buckingham



How to Survive the 21st Century

YUVAL NOAH HARARI

Read Yuval Harari's blistering warning to Davos in full

<https://www.weforum.org/agenda/2020/01/yuval-hararis-warning-davos-speech-future-predications/>



AZ ADATVEZÉRELT GONDOLKODÁS SZINTJEI

PAZARLÓK

GYŰJTÖGETŐK

ADATVEZÉRELTEK

DRIVERLESS AI

H2O.ai

0.0.1

TRAINING DATA

dataset
default_of_credit_card_clients.csv

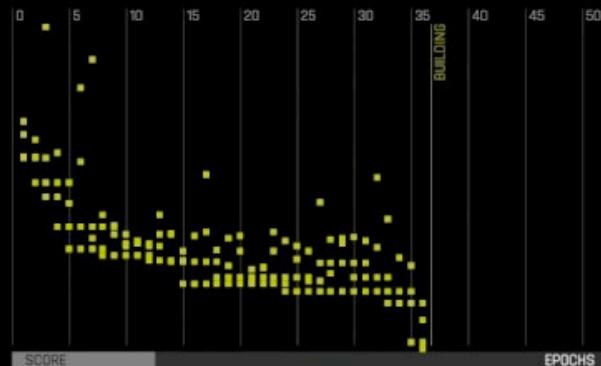
ROWS 24k COLUMNS 25 DROPPED 0 IGNORED 1

TARGET COLUMN

default payment next month

TYPE int64 NA 0 MEAN 0.22 STD DEV 0.41

ITERATION SCORES



STATUS: RUNNING



EXPERIMENT SETTINGS



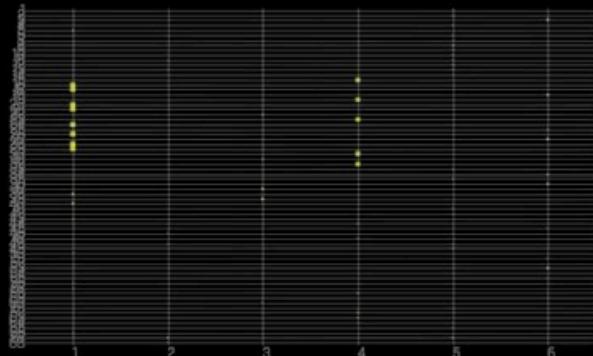
GPU STATS



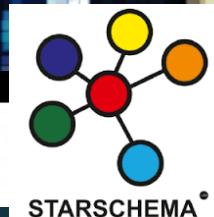
VARIABLE IMPORTANCE

58_Interaction_PAY_0#multiply#BILL_AMT1	14625.96
10_PAY_0	7999.05
41_Interaction_PAY_0#safe_divide#PAY_AMT5	5094.40
43_TruncSVD_PAY_2_PAY_5_0	3201.62
70_PAY_0	3148.27
30_Interaction_BILL_AMT2#safe_divide#LIMIT_BAL	2895.04
66_Interaction_PAY_AMT3#multiply#LIMIT_BAL	2249.62
56_Interaction_BILL_AMT2#multiply#LIMIT_BAL	1583.83
28_TruncSVD_BILL_AMT1_LIMIT_BAL_0	1268.93
71_Interaction_PAY_AMT6#subtract#PAY_AMT4	1132.04
56_Interaction_PAY_0#multiply#BILL_AMT2	1130.92
61_ClusterID_93	1054.19
44_CV_CatNumEnc_PAY_6_PAY_2_std	1015.55
12_PAY_3	1000.46

FEATURE TRANSFORMATIONS



Nokia Siemens
Networks



ERICSSON



Continental

SEQUENCE IQ

General Electric

kaggle™

CRUNCH

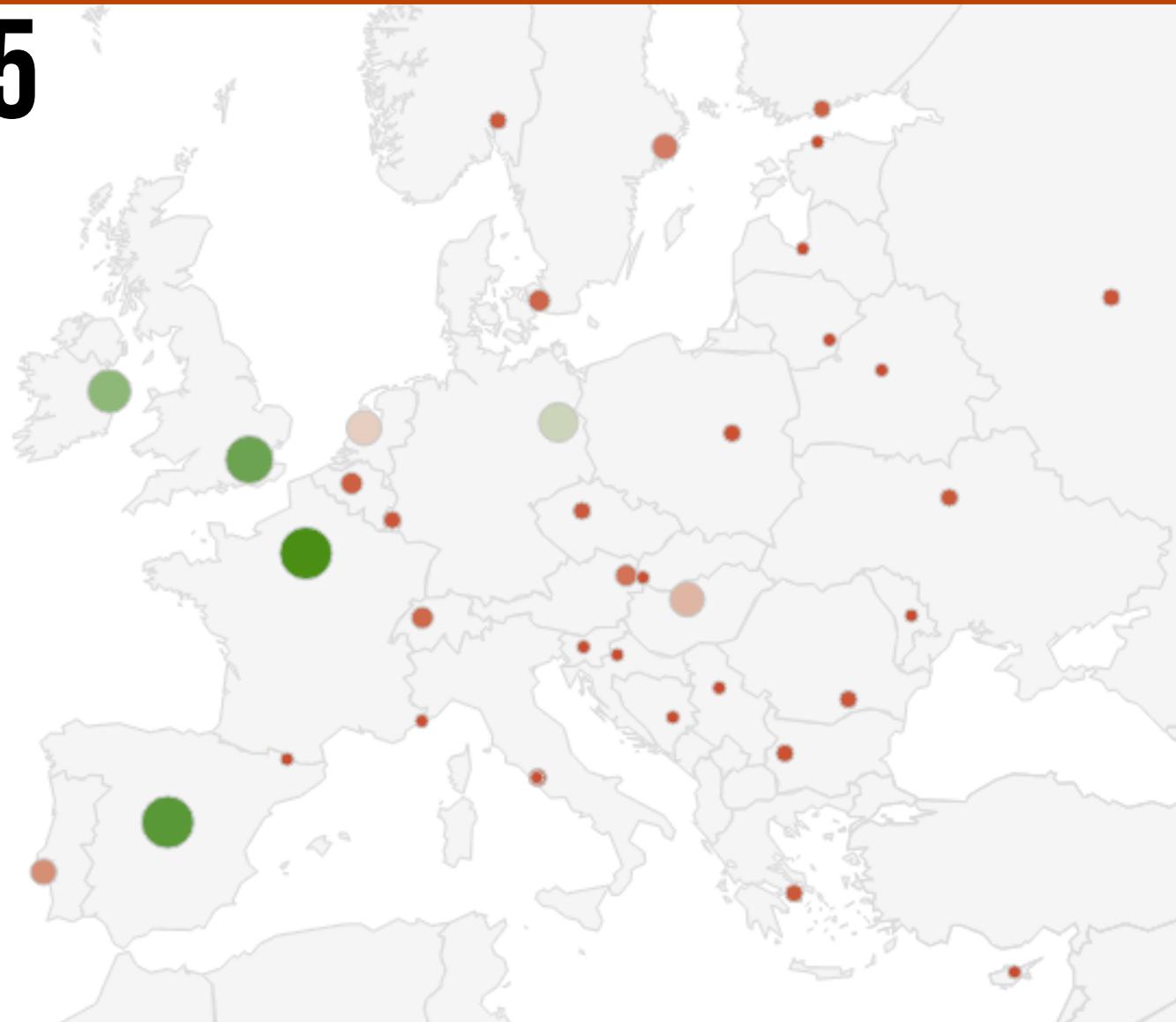
REINFORCE

meetup

BUDAPEST ÉS AZ ADAT EGYMÁSRA TALÁLT

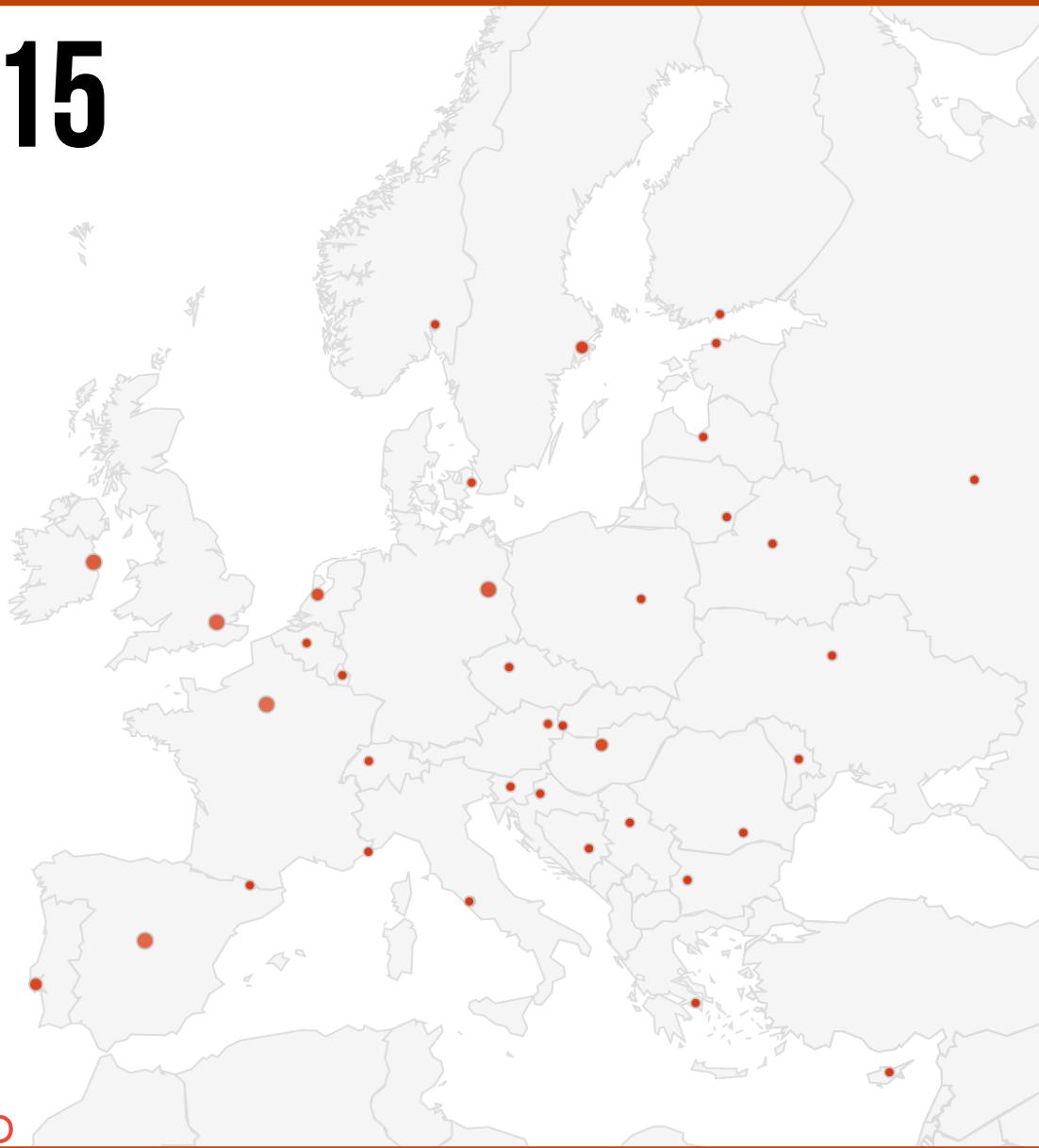
LÉTEZIK-E DATAPEST?

2015



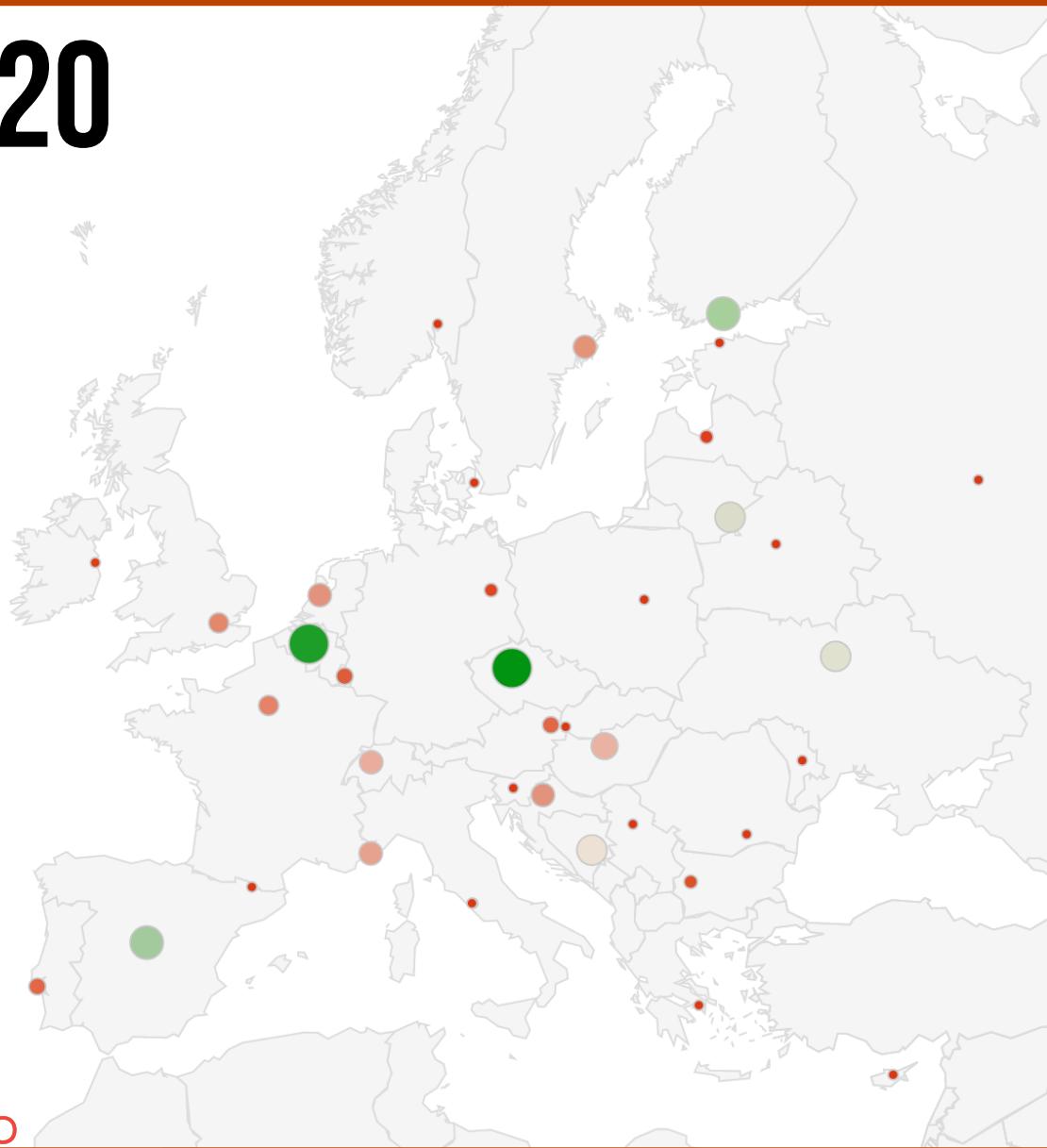
LÉTEZIK-E DATAPEST?

2015

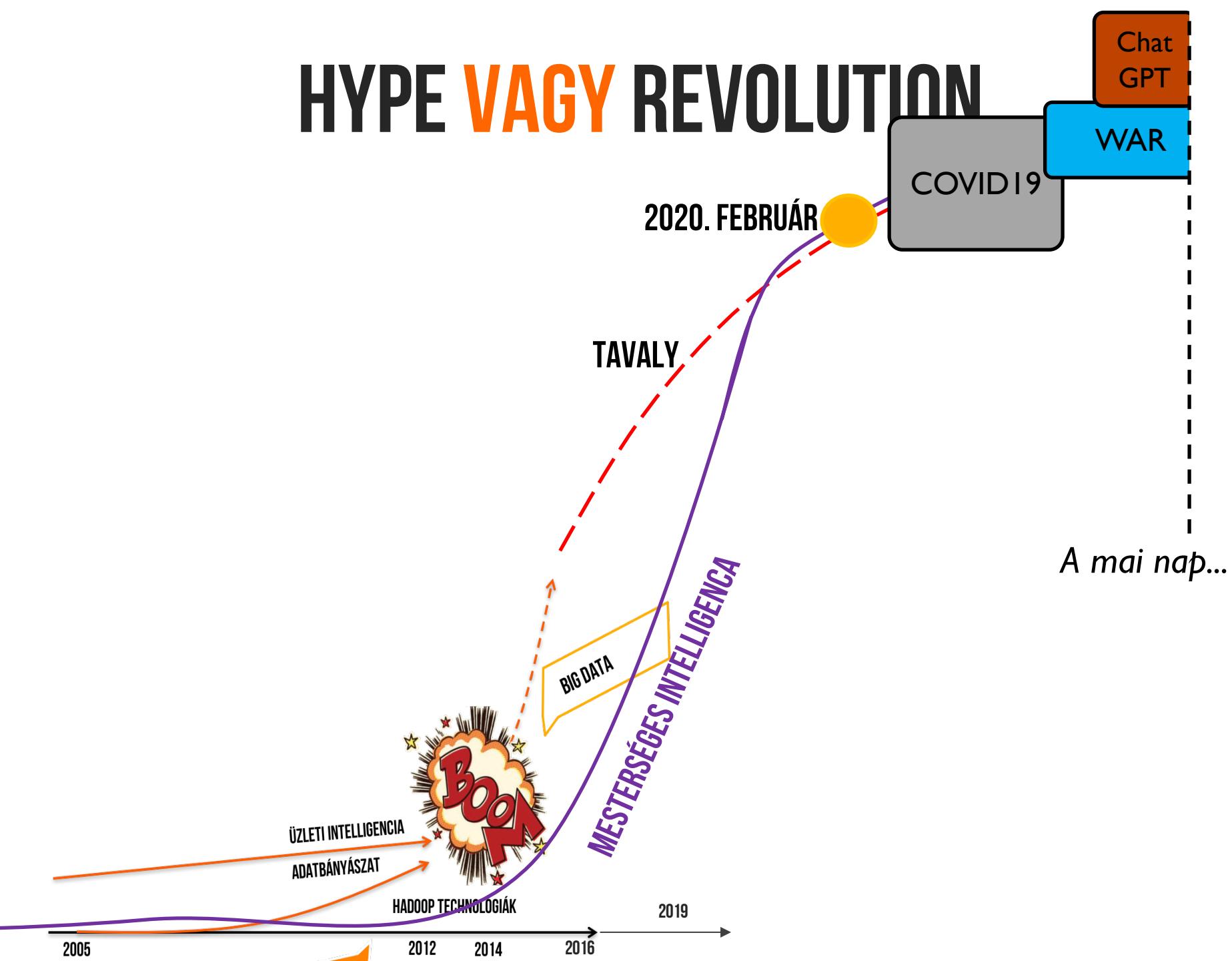


LÉTEZIK-E DATAPEST?

2020



HYPE VAGY REVOLUTION





MERRE TARTUNK? - UTÓPIÁK



VÁLLALATI



ÁLLAMI



EGYÉNI



ARTIFICIAL
INTELLIGENCE

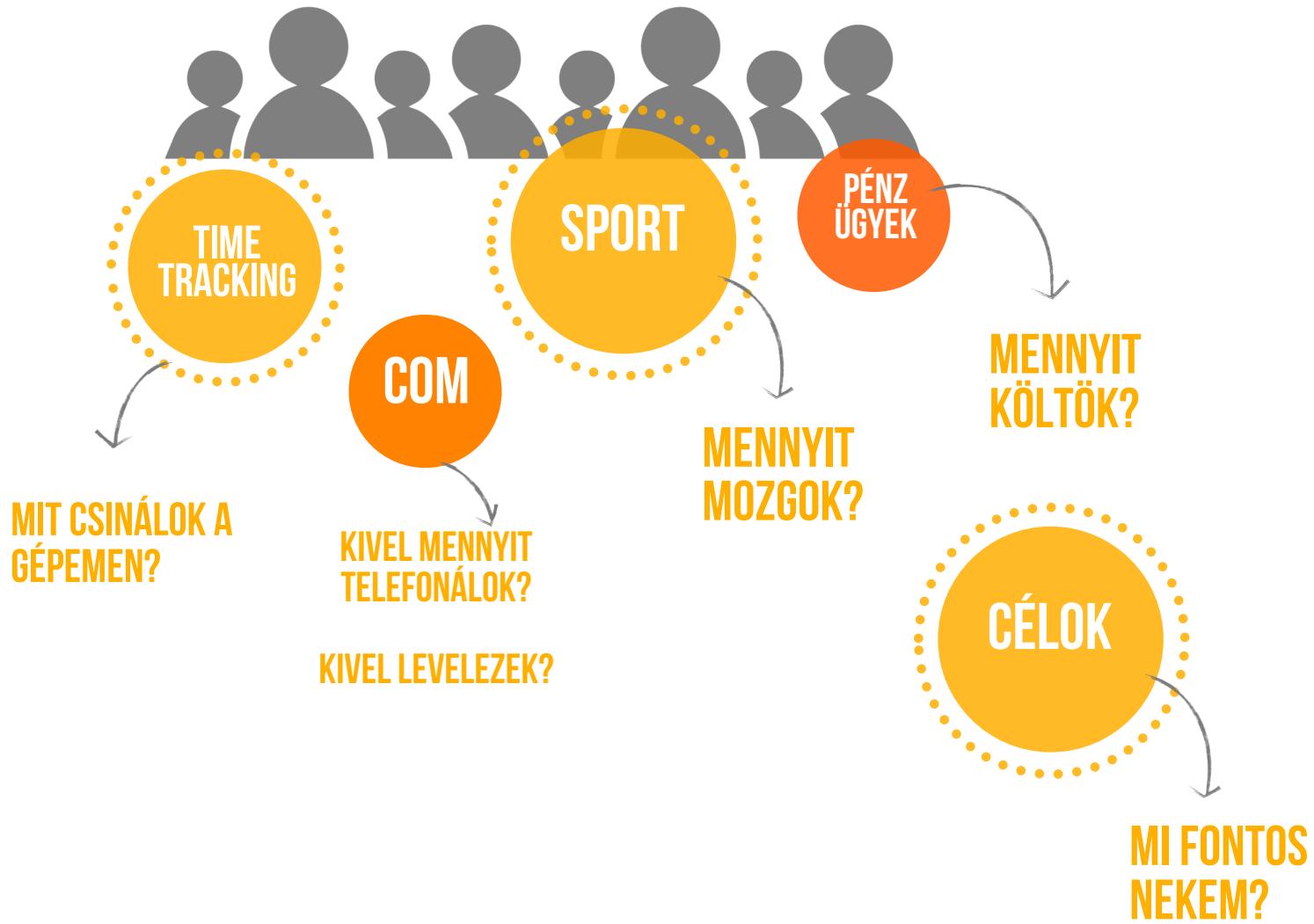


ChatGPT





TUDATOSSÁG VS. “ADATOSSÁG”



KÖSZÖNÖM A FIGYELMET!

Gáspár Csaba

dmlab.hu/blog

gaspar.csaba@tmit.bme.hu
+36 20 8234 154







Adatbányászati technikák

Toka László



Heti bontás

1. Bevezetés
2. Adatbányászati szoftverek áttekintése, piaci áttekintés
3. Adatfelderítés
4. Adatelőkészítés
5. Osztályozók 1
6. Osztályozók 2
7. Húsvéti szünet
8. Osztályozók 3, zh1
9. Osztályozók kombinálása
10. Klaszterezés 1 (**hétfő helyett pénteken**)
11. Klaszterezés 2
12. Anomália detektálás
13. zh2 + házi bemutatása
14. Pünkösdi szünet

2023. 03. 13.

Adatbányászati technikák - 3. előadás

Adathalmaz



- elvileg bármi, ami információt hordoz és amiből valamilyen összefüggéseket akarunk kinyerni
- leggyakrabban úgy gondolunk az adathalmazra, mint egy táblázatra (data frame)
- sorok (rekordok): az egyes megfigyelések, emberek, esetek
- oszlopok az attribútumok, ezek azok a jellemzők, amik valamilyen értéket felvesznek minden egyes sorban
- egy esetet jellemznek a sorának az attribútum-értekei
- lehet az adat eredendően másféle is, de arra törekszünk, hogy ilyen alakra hozzuk



Nyers adat (raw data)

- ahogy az adatot kapjuk, eredeti állapotában
- így nem lehet vele dolgozni, előfeldolgozás (preprocessing) szükséges
- data wrangling/munging: az adatok elfogadható, feldolgozható formára hozása,
- nincs minden bevált recept, sok idő de csak egyszer kell megcsinálni



Feldolgozott adat (processed data)

- feldolgozásra alkalmas állapotba hozott adat
- sok lépésből állhat az előfeldolgozás (erről később részletesen)
- nagyon fontos, hogy az előfeldolgozás is dokumentáltan történjen (honnan töltöttem le az adatot, mit csináltam vele, használt kódok is)



Elvárások

- egy táblában (egy sorhalmazban) azonos típusú sorok legyenek csak: pl. csak kórházak statisztikái vagy csak egyes emberekre vonatkozó sorok
- egy sor egy esetnek feleljen meg (pl. egy kórház vagy egy ember, egy eset)
- egy oszlop egy változónak feleljen meg, konzisztensen



További elvárások

- Jó lenne tudni (a valódi munka elkezdése előtt), hogy
 - melyik oszlop milyen típusú adatot tartalmaz: attribútum fajtája, jelentése
 - vannak-e hiányzó értékek
 - vannak-e kilógó értékek (outlier)
 - attribútum-értékek eloszlása milyen az egyes oszlopokon belül: át kell-e skálázni valamit
 - van-e redundancia, azaz vannak-e azonos információt hordozó oszlopok
- Ennek eléréséhez mindenféle technikák vannak, erről majd az adatelemzés felépítésénél beszélünk részletesebben



Attribútumok típusai: egy lehetséges felosztás

- **folytonos:**
 - valós értékeket vesz fel (de néha azt is folytonosnak hívjuk, amikor megszámlálhatóan végtelen lehetséges érték van)
 - pl. hőmérséklet, magasság, testsúly
- **diszkrét:**
 - véges sok (vagy megszámlálhatóan végtelen sok érték)
 - pl. irányítószám, életkor, nem, darabszám
 - gyakran egész számokkal reprezentált, néha címkékkel (label)
- **bináris:**
 - speciális diszkrét attribútum: 0 és 1 a lehetséges értékek
 - gyakran aszimmetrikus jelentésű: a 0 azt jelenti, hogy valami nincs, nem igaz
 - gyakran ritka adatmátrixokban szerepel: nagyon sok a 0 (például dokumentum-szó mátrixok)
 - speciális kezelés lehet néha szükséges



Attribútumok típusai: egy hasonló felosztás

- kvalitatív attribútumok (categorical/nominal attribute)
 - címkék, például személy neme, családi állapota, kapott terápia, túlsúlyos-e?
 - értelmes műveletek: gyakoriságok (hisztorogramon ábrázolva)
 - jó, hasznos, ha az attribútumok értékei kifejezők (pl. férfi-nő jobb, mint az 1-2)
- kvantitatív attribútumok (numerical attribute)
 - életkor, testsúly, testmagasság, BMI index
 - értelmes műveletek: medián, percentilisek, esetleg átlag, szórás
 - kérdés, hogy csak a sorrend számít vagy a különbség illetve az arány is értelmes, pl. 20 °C az nem kétszer olyan meleg, mint 10 °C



Attribútumok típusai: még egy felosztás

- Rekord típusú adatokból álló táblázat, mátrix
 - számokból álló m soros, n oszlopos táblázat
 - gyakran az n dimenziós tér pontjainak tekintjük a sorokat
 - speciális eset: dokumentum-szó mátrix
 - sorok a dokumentumok, oszlopok a kulcsszavak
 - bináris attribútum mutatja, hogy szerepel-e az adott szó vagy diszkrét attribútum mutatja az előfordulás darabszámát
 - általában rengeteg oszlop van, nagy a dimenzió
 - speciális eset még: tranzakciós adatokból származtatott adatmátrix: eredetileg halmazok, de könnyen átalakítható a dokumentum-szó mátrixhoz hasonlóan



Attribútumok típusai: még egy felosztás

- Nem rekord típusú adathalmaz, ilyeneket általában addig alakítjuk, amíg rekord típusúak lesznek
 - grafikus adatok: molekulák közötti kapcsolatok, pl. ki kivel kapcsolódik, kötések szögei
 - képek: pixelsorozatra fordítható le vagy valami származtatott feature-lista alapján kap számszerűsíthető attribútumokat minden kép
 - térbeli és/vagy időbeli kapcsolat is van a sorok között: pl. adott pillanatban meteorológiai mérések több helyen (ábrázolásnál jó ennek tudatában lenni)





“

Adatminőség





Adatminőséggel kapcsolatos kérdések

- Mik a lehetséges problémák az adattal?
- Hogy vesszük észre ezeket?
- Hogyan kezeljük a megtalált hibákat?



Mik a lehetséges problémák az adattal?

- mérési hibák
- inkonzisztencia, pl. az adathalmaz egyik felében km, a másikban m-ben vannak az adatok
- hiányzó adatok
- duplikátumok: feleslegesen ismétlődő sorok, nem minden teljesen egyformák, pl. adatbázisban ugyanaz az ember több hasonló lakcímmel
- furcsa, nehezen hihető adatok (mindenki túlsúlyos az adatbázis szerint vagy minden lakásban 100-nál több szoba van)
- outlier-ek: kilógó, furcsa, másmilyen sorok vagy attribútumértékek (lehet, hogy baj, lehet, hogy nem)



Hogy vesszük észre ezeket?

- ez az előfeldolgozás és a felderítés (exploratory data analysis) része
- grafikus ábrázolás: eloszlások, hisztogramok
- összegző függvények futtatása az adatokra (mean, median, percentilisek)
- python pandas-ban head(), info(), describe(), corr(), stb.



Hogyan kezeljük a megtalált hibákat?

- az minden jó, ha legalább tudjuk, hogy mivel állunk szemben
- van amivel nem lehet sokat tenni (pl. mérési hiba), de legalább tudatában vagyunk annak, hogy volt ilyen
- amúgy meg adattisztítás, erről később részletesen
- hiányzó értékek:
 - lehet, hogy nem baj (nem minden sorban értelmes az adott attribútum)
 - megoldás lehet az adott érték pótlása vagy a sor törlése
 - az is lehet, hogy elég, ha tudunk a jelenségről
- duplikátumok: észrevenni őket és azonosítani a közel azonosakat (néha csak ezt a részt hívjuk adattisztításnak)
- outlier: lehet, hogy el kell hagyni, de lehet, hogy épp az ilyeneket akarom megtalálni



Hasonlóság,
különbség

Bevezetés

- Sokszor fontos lehet annak mérése, számszerűsítése, hogy két sor (két pont) mennyire hasonlít
- Legfontosabb ilyen helyzet a klaszterezés, amikor a hasonlókat akarjuk egybe gyűjteni
- A hasonlóság illetve különbözőség mérésére többféle lehetséges függvény van
- A használt függvény mindenkorban függ attól, hogy milyen típusú attribútumokból áll a sor (folytonos vagy sem, illetve kvalitatív vagy kvantitatív)
- Alapmegközelítés, hogy oszloponként (mezőnként) definiáljuk a távolságot és aztán a sorok távolsága ezekből adódik (erről később)
- Először azt kell tisztázni, hogy egy oszlopon belül mit jelent két érték távolsága

Hasonlóság jellemzői (similarity)

- Azt méri, hogy mennyire hasonlóak, egyformák
- Minél nagyobb a szám, annál hasonlóbbak
- Szimmetrikus, azaz p és q hasonlósága ugyanaz, mint q és p hasonlósága
- Általában $[0, 1]$ közötti értékek (ritkábban $[0, \inf]$ közötti értékeket vesz fel)



Különbözőség (dissimilarity)

- Azt méri, hogy mennyire különböznek
- Minél kisebb az érték, annál egyformábbak
- Általában a 0 jelentése az, hogy egyformák
- Szimmetrikus, azaz p és q különbözősége ugyanaz, mint q és p különbözősége



Mikor mit használunk? Kategorikus attribútumoknál

- hasonlóság: 1, ha egyformák és 0, ha nem egyformák
- különbözőség pont fordítva: 0, ha egyformák és 1, ha nem egyformák
- ha a címkék által kódolt dolgok között van valami csoportosítás, akkor lehet nem bináris is a függvény: aminosav szekvenciák összevetésénél nem csak az számít, hogy egyformák-e, mert vannak nem egyforma, de hasonló aminosavak (hidrofób vs. hidrofil, alakjuk, stb.)
- bioinformatikában rengeteg féle pontozómátrix van: egyforma aminosavakra az érték 0, különben meg minél különbözőbbek, annál nagyobb

Mikor mit használunk? Ha az értékek egy adott intervallumból kerülhetnek ki

- Ha a lehetséges értékek 1, 2, ..., n:
 - különbözőség:
 - p és q különbözősége $|p-q| / (n-1)$
 - ez 0 és 1 közé lövi be a különbözőséget
 - 0, ha megegyeznek
 - hasonlóság:
 - p és q hasonlósága $1 - |p-q| / (n-1)$
 - ez 0 és 1 közé lövi be a hasonlóságot
 - 1, ha megegyeznek

Mikor mit használunk? Ha az értékek nem egy véges intervallumból valók

- különbség:
 - p és q Különbsége $d(p, q) = |p - q|$
 - ez 0 és inf közé lövi be a különbséget
 - 0, ha megegyeznek
- hasonlóság:
- sokféleképpen származtatható a fenti különbségből hasonlóság
- ellentett, azaz $-d(p,q)$: -inf és 0 közötti értékeket vesz fel
- $1 / (1+d)$: 0 és 1 közötti értékek

Több azonos típusú attribútummal rendelkező sor összehasonlítása

- Oszloponként képezzük a távolságot
- Aztán:
 - vagy összegezzük az oszloponkénti távolságokat
 - vagy az összeget elosztjuk az oszlopszámmal
 - vagy súlyozott összeget számolunk (és utána osztunk az oszlopszámmal)
 - oszloponkénti távolságképzés előtt szükség lehet átskálázásra
(standardizálás): azonos nagyságrendűek legyenek az attribútumok értékei
(szobaszám vs. négyzetméter)



Távolság fogalma

- Leggyakrabban egy speciális alakú különbözőség-fogalommal dolgozunk, ennek neve távolság.
- Jellemzői:
 - $d(p, q) \geq 0$ minden igaz és $d(p, q) = 0$ csak akkor, ha $p = q$ (reflexivitás)
 - $d(p, q) = d(q, p)$ (szimmetria)
 - $d(p, q) \leq d(p, r) + d(r, q)$ minden p, q, r esetén (háromszög egyenlőtlenség)
- Más néven: metrika.

Euklideszi távolság

Leggyakrabban ezt használjuk, ha a sorok értelmezhetők n -dimenziós térben levő pontokként

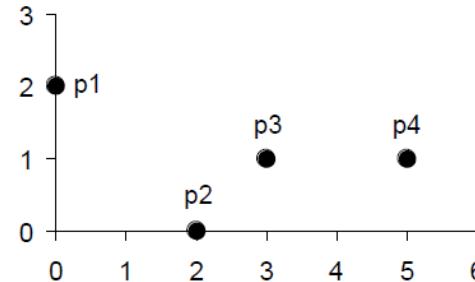
$p = (p_1, \dots, p_n)$ és $q = (q_1, \dots, q_n)$ két pont a térben

$$d(p, q) = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

itt is kellhet előbb a standardizálás:

- $\frac{p - \text{mean}(p)}{\text{sd}(p)}$, azaz kivonjuk az átlagot és osztunk a szórással
- vagy $\frac{p - \text{min}(p)}{\text{max}(p) - \text{min}(p)}$

Euklideszi távolság



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Minkowski távolság

Euklideszi távolság általánosítása

$p = (p_1, \dots, p_n)$ és $q = (q_1, \dots, q_n)$ most is két pont a térben van egy paramétere, r , ez valami $1, 2, \dots$ egész szám

$$d(p, q) = \sqrt{\sum_{k=1}^n |p_k - q_k|^r}$$

$r = 2$ az Euklideszi távolság

itt is kellhet előbb a standardizálás (minél nagyobb az r , annál inkább) ez minden r egész szám esetén metrika

Minkowski távolság

$r = 1$: Manhattan távolság

- L_1 távolsága $(1, 2)$ és $(7, 0)$ -nak 8, ennyi blokkra/sarokra vannak egymástól

$r = 2$ az Euklideszi távolság

van olyan is, hogy $r = \infty$, ez az L_∞ , néha hívják L_{max} -nak is

- egyik definíció: $d(p, q) = \lim_{r \rightarrow \infty} \sqrt[r]{\sum_{k=1}^n |p_k - q_k|^r}$
- ami ugyanaz, mint $d(p, q) = \max_{k \in \{1, 2, \dots, n\}} |p_k - q_k|$
- ez is metrika

Minkowski távolság

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

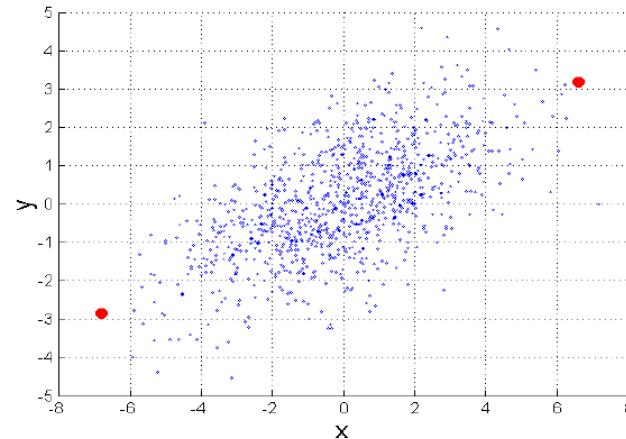
L ∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Mahalanobis távolság

- a Minkowski távolságok nem veszik figyelembe, hogy az adatmátrix oszlopai nem feltétlenül függetlenek
- szélsőséges esetben lehet két azonos oszlop, ennek eltérése így duplán számít
- erre megoldás lehet az, ha a mátrixot átalakítjuk az elemzés előtt, új változók bevezetésével vagy a régiek közül néhány elhagyásával (erről később részletesen lesz szó)
- vagy megoldás az, ha olyan távolságfogalmat használunk, ami ellensúlyozza az oszlopok korreláltságából adódó torzítást

Mahalanobis távolság

$$mahalanobis(p, q) = (p - q) \Sigma^{-1} (p - q)^T$$

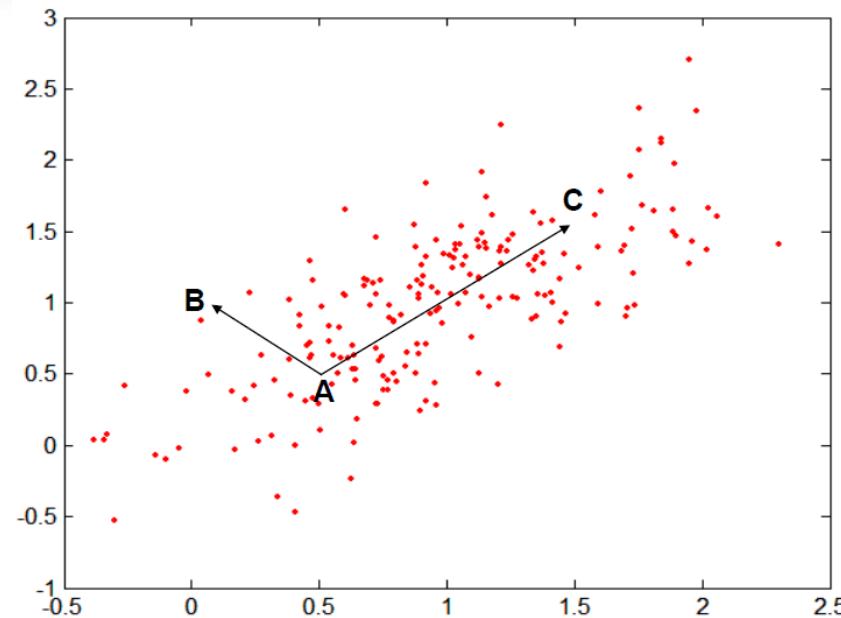


Σ is the covariance matrix of the input data X

$$\Sigma_{j,k} = \frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)(X_{ik} - \bar{X}_k)$$

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

Mahalanobis távolság



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

Bináris vektorok távolsága

- ha binárisak az adatok, akkor nagyon gyakran ritka adatmátrixról van szó: szinte minden bejegyzés 0 (dokumentum-szó mátrix, tranzakciós mátrix)
- ebben az esetben az eddigi távolságfogalmak nem informatívak: szinte mindenki egyformának látszik
- kéne valami speciálisabb távolság ezekre az esetekre
- p és q most is n hosszú vektorok, de minden komponens értéke 0 vagy 1
- itt hasonlóságok vannak (azaz minél nagyobb az érték, annál egyformábbak)



Simple matching coefficient (SMC)

- M_{01} = hány helyen van p-ben 0 és q-ban 1
- M_{10} = hány helyen van p-ben 1 és q-ban 0
- M_{00} = hány helyen van p-ben és q-ban is 0
- M_{11} = hány helyen van p-ben és q-ban is 1
- $SMC = (M_{00} + M_{11}) / (M_{00} + M_{11} + M_{01} + M_{10})$
- SMC tehát = ahol egyeznek osztva az attribútumok számával
- ez lényegében az L1 távolságnak megfelelő hasonlóság

Jaccard együttható

- SMC nem jól mér, ha ritka az adatmátrix mert nagyon befolyásolja a SMC szerinti hasonlóságot ha sok közös nulla van (pl. sok olyan szó, ami egyik dokumentumban sincs benne)
- megoldás: a közös nullák ne számítsanak: Jaccard együttható
- $$\text{Jaccard} = M_{11} / (M_{11} + M_{01} + M_{10})$$
- hány közös előfordulás van a valahol elforduló szavak számához képest



Jaccard együttható

$p = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$q = 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

$$\text{SMC} = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7) / (2+1+0+7) = 0.7$$

$$J = (M_{11}) / (M_{01} + M_{10} + M_{11}) = 0 / (2 + 1 + 0) = 0$$

Cosine hasonlóság

- dokumentum-szó mátrix esetén hasznos, ha a mátrix gyakoriságokat tartalmaz (nem bináris, hanem azt mutatja, hogy hányszor szerepelt egy kulcsszó)
- p és q két azonos hosszúságú, egész számokból álló vektor (továbbra is igaz, hogy sok bennük a nulla)
- $\cos(p, q) = p \cdot q / (\|p\| \|q\|)$
- azaz skalárisan összeszorozzuk a két vektort és osztunk a hosszuk szorzatával
- ismert középiskolából, hogy ez a síkon a két vektor szögének a cosinus-a
- hasonló több dimenzióban is

Cosine hasonlóság

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \|d_2\| ,$$

where \bullet indicates vector dot product and $\|d\|$ is the length of vector d .

- Example:

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

$$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|d_1\| = (3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$$

$$\|d_2\| = (1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2)^{0.5} = (6)^{0.5} = 2.245$$

$$\cos(d_1, d_2) = .3150$$

Különböző fajta attribútumok összehasonlítása

- az eddigi módszerek akkor jók, ha az összehasonlítandó vektorok azonos típusú értékeket tartalmaznak minden oszlopban
- ha nem így van:
 - csoportosítsuk össze az egyformákat: binárisak, kategorikusak, folytonosak, stb.
 - számoljuk ki az egyes csoportokra a hasonlóságot vagy távolságot
 - arra figyeljünk, hogy azonos típusú dolgot számolunk mindenhol (vagy távolság vagy hasonlóság)
 - valahogyan (esetleg súlyozva az egyes részek nagysága vagy értéke szerint) eredő távolságot vagy hasonlóságot definiálunk
 - akkor is akarhatunk súlyozni, ha egyszerűen csak vannak attribútumok, amik kevésbé fontosak

például L_r normát is lehet súlyozni:

$$\sqrt{\sum_{k=1}^n w_k \cdot |p_k - q_k|^r}$$

Korreláció

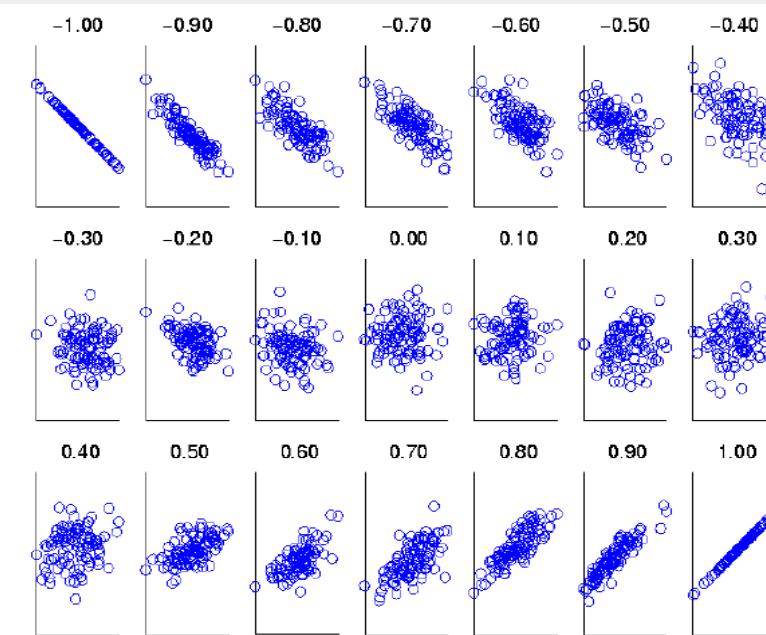
- ezzel általában oszlopokat hasonlítunk össze
- nem az algoritmusokban használjuk, hanem az előfeldolgozásnál, amikor az algoritmusokban használt attribútumokat határozzuk meg
- két oszlop, azaz két attribútum közötti lineáris kapcsolatot méri
- arra lehet jó, hogy ha nagy a korreláció két oszlop között, akkor esetleg elég egyiket bevenni az elemzésbe
- vigyázat! nem minden kapcsolatot derít fel, csak a lineárisat!

előbb standardizáljuk az oszlopokat: p_k helyett $p'_k = \frac{p_k - \text{mean}(p)}{\text{sd}(p)}$,

hasonlóan q'

$\text{correlation}(p,q) = p' \cdot q'$ (skalárszorzat)

Korreláció





Gyakorlat

- A $p = (1, 1, 1, 0, 0, 0)$ és $q = (0, 0, 1, 1, 1, 0)$ vektorokra határozza meg az L_1 , L_2 , L_{\inf} távolságot, az SMC-t és a Jaccard-hasonlóságot.
- A $p = (2, 1, 0, 7)$ és $q = (1, 3, 0, 0)$ vektorokra határozza meg az L_1 , L_2 , L_{\inf} távolságot, és a cosine-hasonlóságot.
- Google colab-ban töltse be a sample_data folderben lévő california_housing_train.csv adatszettet, és keresse meg a 2 legnagyobb korrelációt mutató attribútumot.



Adatbányászati technikák

Toka László



Heti bontás

1. Bevezetés
2. Adatbányászati szoftverek áttekintése, piaci áttekintés
3. Adatfelderítés
4. Adatelőkészítés
5. Osztályozók 1
6. Osztályozók 2
7. Húsvéti szünet
8. Osztályozók 3, zh1
9. Osztályozók kombinálása
10. Klaszterezés 1 (hétfő helyett pénteken)
11. Klaszterezés 2
12. Anomália detektálás
13. zh2 + házi bemutatása
14. Pünkösdi szünet

2023. 03. 20.

Adatbányászati technikák - 4. előadás

Mivel kezdődik az adatbányászat?



- majd tanulunk konkrét eljárásokat, amikkel az adatokból mindenféle érdekes infó nyerhető ki
- de ahoz, hogy ezek menjenek szép adatok kellenek
- eredendően az adat sose szép, valamit biztos csinálni kell vele
- ez sok munka, nem egzakt feladat
- de azért a fő részeire van egy protokoll



Honnan szerzünk adatokat?

- néha úgy találjuk készen, valaki összegyűjtötte (ingyen elérhető, meg kell venni)
- szinte sose pont olyan, mint ami nekünk kell
- sokszor elosztottan van
- esetleg több táblából kell valahogy egyet csinálni (adatbázis kezelés)
- fontos, hogy dokumentáljuk, hogy honnan szereztük, honnan töltöttük le
- ha valaki már előfeldolgozta valahogy, akkor is értelmes látni a nyers adatot vagy legalább megérteni, hogy mi történt a feldolgozás során



Fő részek, ha már megvan az adat

- ismerkedés: milyen típusú attribútumok vannak, mit kódolnak, hogyan (ezt érintettük már a múltkor)
- exploratory elemzés: grafikonok, ábrák, mert így könnyebb látni mintázatokat
- preprocessing: attribútumok, illetve sorok számának csökkentése



Ismerkedés az adattal

- honnan van az adat? hogyan gyűjtötték?
- elévült-e már az adat?
- attribútumok típusa, tipikus értékei, volt-e default érték a bevitelkor
- emlékeztető: python pandas-ban head(), info(), describe(), corr(), stb.





“

Adatfelfedezés



Exploratory elemzés: mi ez?

- ez alapján lehet eldönteni, hogy
 - milyen algoritmust használunk
 - egy adott algoritmusban milyen attribútumok a fontosak (hol lehet érdekes, megvizsgálandó kapcsolat vagy hol van redundancia)
 - látszik-e valami nyilvánvaló hiba vagy tennivaló az adatokkal (átskálázás, hiányzó értékek, kilógó értékek)
- vannak olyan mintázatok, amiket egy jól sikerült ábrán az ember gyorsan felismer



Exploratory elemzés: fő részei

- összegző statisztikák készítése
- ábrázolás



Összegző statisztikák

- ezt már érintettük, amikor az adattal való ismerkedésről volt szó
- célja, hogy valami számszerű adattal összegezzük a változók értekeit
 - gyorsan számolható legyen
 - informatív legyen
- kábé hol vannak az értékek, mennyire szóródnak, mik a gyakoriságok
- általában vannak mindenféle mindenféle hasznos parancsok erre



Összegző statisztikák

- kategória típusú változónál a gyakoriságok informatívak
- percentilisek a folytonos adatokhoz jók
 - 0 és 100 közötti percentilisekről beszélünk
 - egy halmaz (attribútumhalmaz, adott oszlop értékei) p-percentilise az az x_p érték, aminél a halmaz értékeinek $p\%$ -a kisebb egyenlő
 - például $x_{50\%}$ azt az értéket adja meg, aminél az összes előforduló érték fele nem nagyobb
 - szokás nézni a 25, 50, 75 percentilist és a min és a max értéket
 - van függvény, ahol beállítható, hogy milyen percentiliseket akarok, default a 0, 25, 50, 75, 100 (ahol 0 a min érték és 100 a max érték)



Átlag (mean) és medián (median)

- az átlag (az adatok számtani közepe) az egyik leggyakoribb összegző függvény
 - az átlag nagyon érzékeny a kilógó adatokra
 - ezért sokszor a mediánt használjuk helyette
- medián: hasonló az 50%-os percentilis értékéhez, de nem egészen az
 - ez persze nem ugyanaz, mint az átlag
 - az emberek több, mint 99%-ának az átlagnál több lába van

$$\text{median}(x) = \begin{cases} x_{r+1} & \text{if } m = 2r + 1 \\ \frac{1}{2}(x_r + x_{r+1}) & \text{if } m = 2r \end{cases}$$

Szórás-szerűségek

- range: milyen tartományba esnek az adatok (max - min)
- szórásnégyzet illetve szórás: $\frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2$
- de ez is érzékeny a kilógó értékekre, ezért néha inkább
$$\frac{1}{m} \sum_{i=1}^m |x_i - \bar{x}|$$
- ábrázolásnál lesznek majd olyan technikák, amikkal ezeket a mennyiségeket jól lehet látni



Ábrázolás célja az ismerkedés során

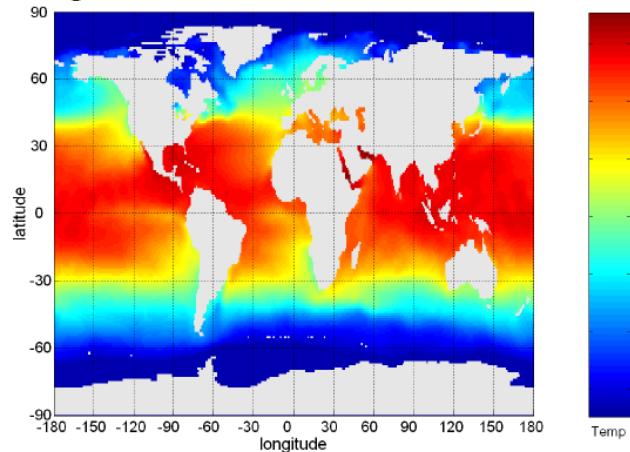
- az adatok közti kapcsolatot vagy adat tulajdonságait mutató jellemzőket ember számára feldolgozható módon megjeleníteni
- ember számára könnyebb egy grafikont értelmezni, mint egy táblázatot
- minták jobban látszanak (ember számára)
- kilógó adatok, furcsaságok is jobban kiugranak
- majd lesz szó arról, hogy az ábrázolás milyen szerepet kap az eredmények ismertetésekor
- általában sok ábra készül, gyorsan



Példa

The following shows the Sea Surface Temperature (SST) for July 1982

- Tens of thousands of data points are summarized in a single figure



Milyen a jó ábrázolás?

- fontos a jó elrendezés
- cél egy jól értelmezhető ábra
- általában nem lehet mindenöt egy ábrában áttekinteni
- ügyesen választunk néhány attribútumot, amiket vagy amiknek a kapcsolatát megvizsgáljuk
- az exploratory elemzésnél elég, ha mi értjük, hogy mi van az ábrán
- sok fajtája lehet, pl. hisztogram, boxplot, scatterplot, stb.

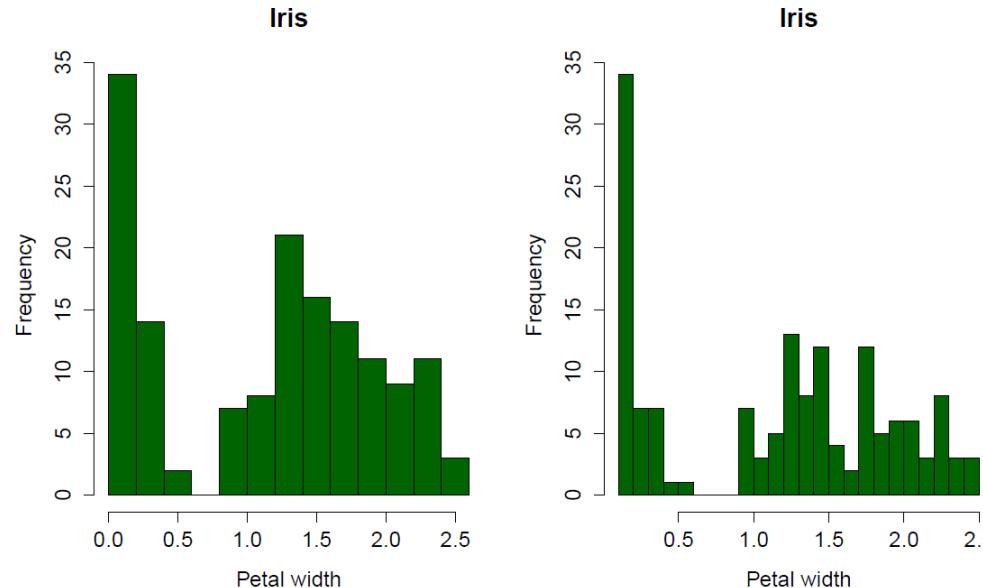


Hisztogram

- egy változó értekéinek eloszlását mutatja
- csoportokba osztja az értékeket és az egy csoportba esők darabszámát mutatja
- az oszlopok magassága a darabszámot jelzi
- működik kategorikus és folytonos attribútumokra is

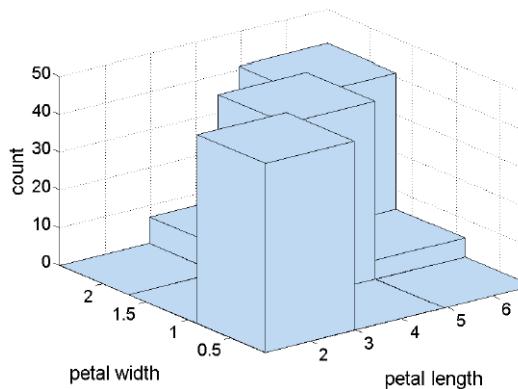


Példa - hisztogram



Példa – 2D hisztogram

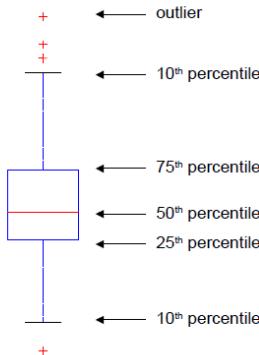
- Show the joint distribution of the values of two attributes
- Example: petal width and petal length
 - What does this tell us?



Példa - boxplot

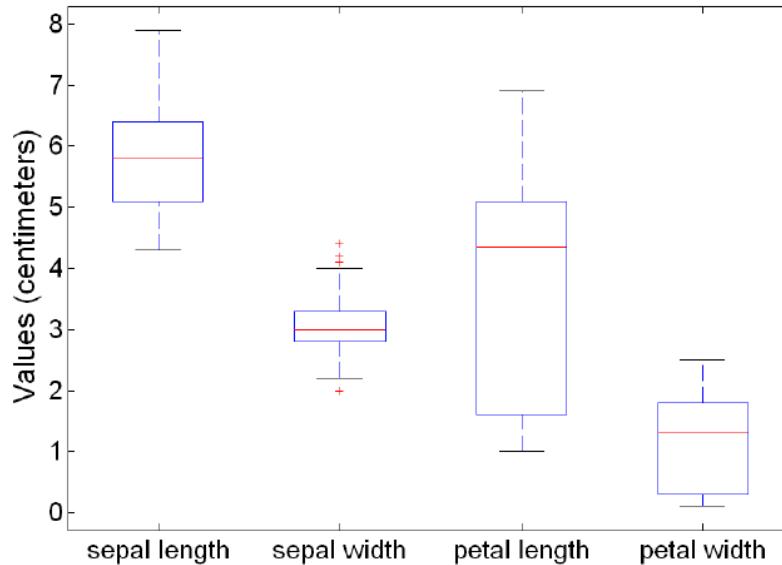
Box Plots

- Invented by J. Tukey
- Another way of displaying the distribution of data
- Following figure shows the basic part of a box plot



Példa - boxplot

Box plots can be used to compare attributes

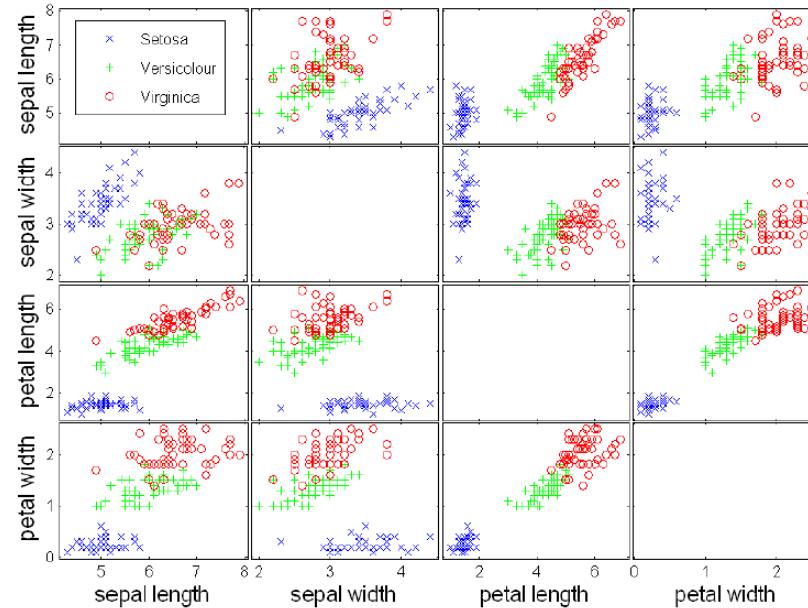


Scatterplot

- soroknak, objektumoknak pontok felelnek meg a síkon vagy esetleg térben
- a pontok helye megfelel a két vagy három kiválasztott attribútum értekéinek
- a max. három kiválasztott dimenzió felül a pontoknak lehet színe és/vagy alakja, és/vagy mérete, ezekkel együtt max. 5-6 dimenzió ábrázolható
- de azért igazából 4 dimenzió felett már nehéz értelmezni, amit látunk



Scatterplot



Az eredmények prezentálása

- az ábrázolás fontos az eredmények prezentálásakor is
- részben hasonló elvek vonatkoznak rá, mint az exploratory ábrázolásra
 - fontos a jó elrendezés, cél a jól értelmezhető ábra
 - a legfontosabb eredményeket kell megmutatni, minden nem lehet
 - sok fajtája lehet, pl. hisztogram, boxplot, scatterplot, stb.
- ami nagyon más: nem elég, ha mi értjük, hogy mi van az ábrán
- értelmes ábracím, tengelyek rendes elnevezése, skála mérete, informatív képaláírás





“

Előfeldolgozás



HSNLab

Cél

- kevesebb oszlop legyen: oszlopok elhagyása, összevonása, új (jobb) feature-ök bevezetése régiek elhagyása mellett
- sorok számának csökkentése, sorok felosztása training és test (és esetleg validation) halmazra
- mindezt azért, hogy
 - gyorsabban fusson le az algoritmus
 - jobb legyen az eredmény (kifejezőbb attribútumok)





Az előfeldolgozás részei

- feature subset selection: oszlopszámot csökkent viszonylag triviális módon
- aggregáció: összevonás, célja az oszlopszám csökkentése
- mintavételezés (sampling): célja a sorok számának csökkentése
- dimenziócsökkentés: kisebb mátrix legyen, oszlopok számának csökkentése, de nem összevonással
- új attribútumok bevezetése: feature creation (de közben csökken az oszlopszám, ennek spec. esete a dimenziócsökkentés)
- diszkretizálás, binárisra átírás: az oszlop típusát változtatja meg
- attribútumok transzformálása máshogyan: skálázás, standardizálás
- Nem feltétlenül ez a sorrend és nem is kell minden.

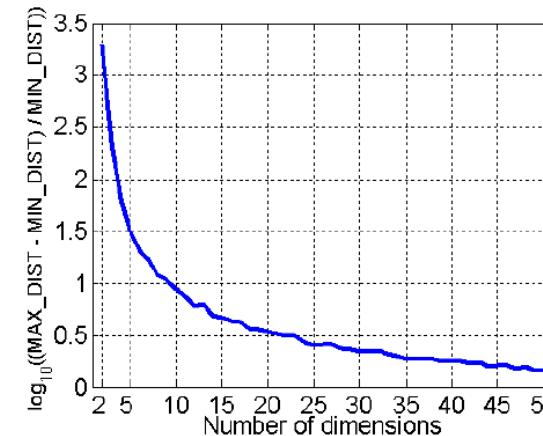


Dimenziócsökkentés: miért?

- ha nagy a dimenzió, akkor
 - lassúak lehetnek az algoritmusok
 - vagy nem is működnek jól
 - meg sok hely is kell az adatok tárolására
- ha kisebb dimenzióban dolgozunk, akkor könnyebb (lehetséges egyáltalán) ábrázolni az adatokat
- tranzakciós és dokumentum mátrixoknál óriási dimenziószám van
- ez azért is baj, mert nagy dimenzióban a pontok közötti eltérések nem különülnek el nagyon
- **Ez a curse of dimensionality.**

Curse of dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points



Dimenziócsökkentés módszerek

- lineáris algebrai módszerek, automatikus
 - a régi attribútumok valami lineáris kompozíciójaként állnak elő az új attribútumok
 - főkomponens analízis: PCA (Principal Component Analysis)
 - szinguláris érték felbontás: SVD (Singular Value Decomposition)
- más módszerek: nem automatizáltak
 - supervised: emberi beavatkozással hozunk létre új változókat, háttértudás birtokában
 - nemlineáris technikák: az új attribútumok a régiekből állnak elő, de nem lineáris kombinációval
- Cél mindenkor az, hogy kevesebb attribútum legyen a végén



Feature subset selection: triviális szűrés

- redundáns oszlopok felismerése
- például eladott termék ára, befizetett ÁFA (amennyiben uaz az áfakulcs minden terméknél, akkor az egyik nem kell)
- irreleváns oszlopok felismerése
- pl. neptun kód irreleváns, ha következő féléves átlagot akarunk előre jelezni
- ha jó dokumentáció van és ismerjük a környezetet, ahonnan az adat jön, akkor ez nem nehéz
- emberi feladat, nem (nagyon) lehet automatizálni



Feature subset selection: alaptechnika

- cél: a triviális szűrés utáni attribútumoknak csak egy részét tartsuk meg
- gyorsabb/jobb legyen az elemzés az új attribútum halmazzal
- futtassuk a használni kívánt adatbányászati algoritmust egy mintán az eredeti és a potenciális szűkebb oszlophalmazzal
- nézzük meg, hogy elromlott-e az eredmény illetve mi történt a sebességgel
- döntsük el, hogy megéri-e a csökkentett attribútumhalmaz



Feature subset selection: módszerek

- brute-force: nézzük meg minden részhalmazát az attribútumhalmaznak: ez nem nagyon járható, már n , az attribútumok száma is nagy, 2^n óriási
- beágyazott módszer: a használt adatbányászati algoritmus majd kiválogatja a fontosokat (pl. döntési fák)
- automatikus szűrés: az algoritmus futása előtt valahogy szűrünk, pl. ha két oszlop korrelációja valami adott értéknél nagyobb, akkor egyiket eldobjuk
- valahogyan (ember?/automatizmus) generálok esélyes részhalmazokat és ezeket tesztelem kis mintán
 - csökkentem egyesével az attribútumok számát, amíg valami STOP-feltétel miatt le nem állok ezzel
 - egy legfontosabb(nak tűnő) attribútummal kezdve egyre többet veszek be, amíg elég jó nem lesz az elemzés



Aggregáció

- valami csoportosítás alapján összegzem a számokat
- ha az adatsorok azt tartalmazzák, hogy melyik város, melyik üzlete, mennyi bevételt produkált egy napon
 - aggregálhatok városra: adott városbeli bevétel egy napon, városok közti összefüggések
 - aggregálhatok időtartamra: boltok havi bevételei, jobban látszanak a boltok közötti sorrendek
- kérdések: mi alapján vonok össze, mit összegez



Aggregáció haszna

- kevesebb sor lesz
- átláthatóbb, esetleg ábrázolhatóbb adatok (kevesebb dimenzió lesz, hatékonyabban lehet ábrázolni)
- stabilabb adatok, tendenciák jobban látszódnak



Mintavételezés

- lehet az adatgyűjtés része is (mikrocenzus)
- az ismerkedéskor is jól jöhet: könnyebben áttekinthető, hogy mivel van dolgunk
- a különböző módszerek tesztelésére elengedhetetlen: nem akarunk minden módszert az egész halmazon lefuttatni
- magában is érdekes lehet, ha túl sok az adat és drága vagy lassú feldolgozni



Mintavételezés alapfeltevései

- olyan minta kell, ami jól reprezentálja a teljes halmazt: reprezentatív
- honnan tudjuk, hogy ilyen-e?
- amikor kábé ugyanaz az eredmény, következtetés, bármi, amiért az egész eljárást csináljuk hasonló a mintán és az egészen
- ez nem valami egzakt
- vannak ennek tesztelésére is technikák (nagy terület)



Mintavételezés típusai

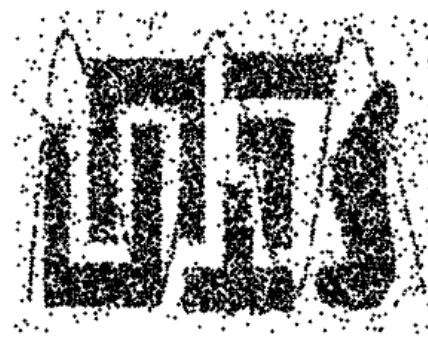
- egy lehetséges felosztás:
- egyenletes eloszlás szerinti random mintavételezés: minden elem ugyanakkora valószínűsséggel kerül be,
 - akkor jó, ha homogén az adatbázis, de ilyenkor sem árt egy permutálás a választás előtt
- több részre osztani a mintát, minden részből választani véletlenszerűen
- visszatevéses-e?



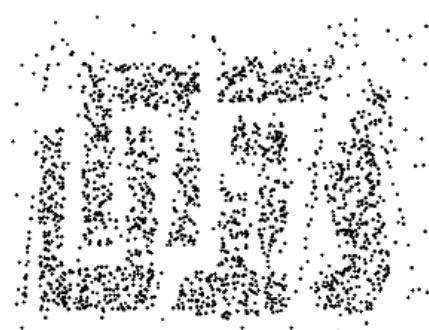
Minta mérete

- nyilván ne legyen nagyon nagy (összemérhető az eredetivel), mert akkor minek csináljuk
- de azért elég nagynak kell lennie ahhoz, hogy jól reprezentáljon
- ha van valami mintázat az adatokon, akkor az látszódjon a mintán is
- egy módszer a progresszív sampling: növelni a minta méretét, amíg az elég jó lesz, pl. predikció minősége szerint

Példa



8000 points



2000 Points



500 Points



PCA és SVD

- mindenkető lineáris algebrai módszer
- vektorok a sorok, eredetileg egy n dimenziós térben
- az egyes oszlopok a dimenzióknak felelnek meg
- a cél egy olyan koordináta-rendszert találni valami alacsonyabb dimenzióban, amire levetítve a vektorokat (azaz sorokat) kevés az információvesztés
- ennek az alacsonyabb koordináta-rendszereknek a vektorai lesznek az új attribútumok
- így kisebb helyen elférnek az adatok (bár információvesztés van)
- felgyorsíthatja az algoritmusokat, ha kevesebb a paraméter



PCA és SVD

- a kovariancia mátrix sajátvektorait keressük meg (ennek minden oszlopszámnyi sajátvektora), ezek lesznek az új attribútumok
- az új dimenzió az lesz, hogy ezeket sajátérték alapján csökkenő sorrendbe téve hányat választok belőlük
- általában ez lassú, ha nagy a mátrix, de utána jól használható kisebb mátrix jön létre
- SVD hasonló céllal, kicsit más módszerrel talál hasonló tulajdonságú vektorokat



Új attribútumok bevezetése

- nem feltétlenül kevesebb attribútum létrehozása a cél
- általános cél: olyan új attribútumhalmazt találni, ami jobban használható
- sokszor (mindig?) emberi feladat, háttértudás kell hozzá
- fajtái:
 - feature extraction: pl. képfeldolgozásnál a pixelek adatait tartalmazó nyers adatból: van-e rajta ember, van-e ilyen vagy olyan kontúr, stb. ehhez ember, vagy ember alkotta spéci algoritmus kell
 - attribútumok kombinálása háttértudással: tömeg és térfogat helyett sűrűsséggel dolgozni



Diszkretizálás

- Célja: folytonos változót diszkrétté alakítani
- ez kellhet, ha
 - olyan algoritmust akarunk futtatni, amihez diszkrét értékű változók kellenek, pl. asszociációs szabályok kutatása, bizonyos típusú döntési fák készítése
 - nem akarunk sok értéket nyilvántartani csak a nagyobb kategóriák a fontosak: magas, közepes, alacsony értékek
- minden értéket valami kategóriába akarunk sorolni
- lehetnek diszjunkt vagy átfedő kategóriák (felhasználástól függően)
- kérdés, hogy hogyan alaktjuk ki a csoportokat



Diszkretizálás: hogyan?

- Kérdés, hogy mire kell a diszkretizálás:
 - ha az exploratory elemzés része (más-e a tendencia alacsony és magas értékek körében), akkor nem érdemes nagyon szofisztikált módszert használni
 - ha a diszkretizálásra alapozunk valami algoritmust, akkor fontos jól csinálni
- Általában jól jön az adatok hátterének ismerete, valami szakértő véleménye



Diszkretizálás: hogyan?

- egyenlő darabszámú csoportokat létrehozva (általában nem jó)
- a folytonos változó értékészletét egyenletesen felosztva csoportostani az elemeket (ez se biztos, hogy jó)
- lehet klaszterezni és a klaszterek azonosítói lesznek a diszkrét változó lehetséges értékei (jobb, de macerás: sok idő, klaszter számot nem ismerjük mindig)



Binárissá átírás

- A diszkretizálás után jön, előbb diszkrét értékű változót kell létrehozni
- asszociációs szabályokhoz elengedhetetlen
- módszere: minden lehetséges diszkrét értékre egy változó, ami vagy
- igaz vagy hamis lehet
- így egy k lehetséges értékű diszkrét változóhoz k új bináris változót kell legyártani
- az i. változó értéke pontosan akkor 1, ha az adott sorban az eredeti változó értéke i volt



Attribútumok transzformálása

- Amikor már minden szép, az adatok rendben vannak, csak az a baj, hogy
 - nem tudjuk jól ábrázolni, mert pl. vannak outlierek, amik miatt az ábra nagyon deformált lesz
 - nem azonos skálán vannak az oszlopok: gyerekek száma vs. fizetés forintban
- Valami bijektív függvényt alkalmazunk: log, kivonás, osztás (normalizálás speciális eset).

Gyakorlat

- vizualizálás
- uni- and bivariate plotting with pandas, plotting with seaborn:
<https://www.kaggle.com/code/residentmario/welcome-to-data-visualization>

Osztályozás

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2018. február 19.

Osztályozás, classification

- adott egy rekordokból álló halmaz, a rekordoknak attribútumaik vannak
- az egyik attribútum a célváltozó, ez kategorikus attribútum, ez reprezentálja, hogy melyik osztályba tartozik az adott rekord
- cél, hogy egy olyan modellt építsünk fel, ami képes megjósolni a célváltozó értékét, ha a többi attribútum értéke adott

Példák

- egy beteg paraméterei alapján eldönten, hogy jó- vagy rosszindulatú daganata van-e
- bankkártyás tranzakció adatait vizsgálva eldönten, hogy van-e csalás vagy nincs (szabályos-e a tranzakció)
- adóbevallásban szereplő értékek alapján megtippelni, hogy gyanús-e
- spam-szűrés

Általános módszer

- van egy csomó rekordunk, ahol minden érték (a célváltozó értéke is) ismert
- ez alapján egy modellt építünk, amit majd olyan új rekordokon használunk, amiknél a célváltozó értéke nem ismert, de minden más attribútumot tudunk
- az ismert rekordokat két részre osztjuk: training set és test set
- a training set-en betanítunk valami modellt
- a test set-en lemérjük, hogy mennyire jó
- ezután használhatjuk élesben

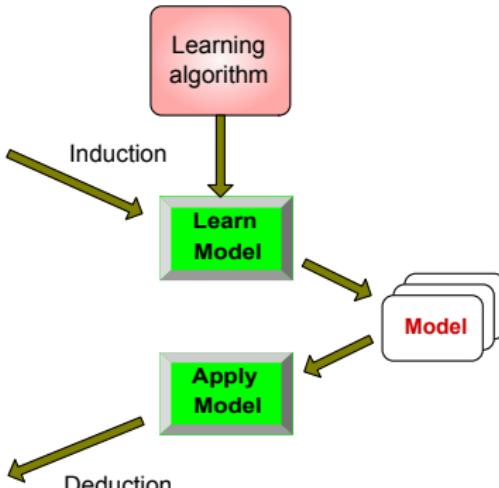
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Kérdések

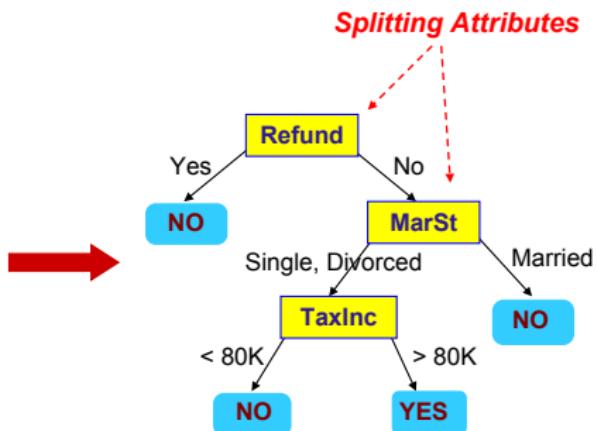
- Milyen modellek vannak?
 - Döntési fák
 - Bayes-osztályozók
 - Mesterséges neurális hálózatok
 - (SVM: Support Vector Machine)
- hogyan állítunk elő egy konkrét modellt? Pl. ha már tudjuk, hogy döntési fát csinálunk, akkor hogyan csináljuk meg; vagy egy ANN-nél hogyan állítjuk be a paramétereket?
- Hogyan mérjük az előállított modell jóságát?
 - accuracy: eltalált címkék száma osztva az összes sor számával
 - error-rate: hibás predikciók száma osztva az összes sor számával

Döntési fa definíció

- gyökeres, lefelé irányított (legtöbbször) bináris fa
- belső csúcsok változókkal és ezekhez kapcsolódó feltételekkel címkézettek
- levelek a célváltozó valamely értékével címkézettek

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat	class
1	Yes	Single	125K	No	
2	No	Married	100K	No	
3	No	Single	70K	No	
4	Yes	Married	120K	No	
5	No	Divorced	95K	Yes	
6	No	Married	60K	No	
7	Yes	Divorced	220K	No	
8	No	Single	85K	Yes	
9	No	Married	75K	No	
10	No	Single	90K	Yes	

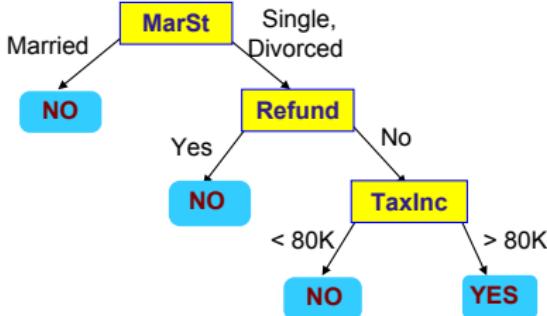


Training Data

Model: Decision Tree

Another Example of Decision Tree

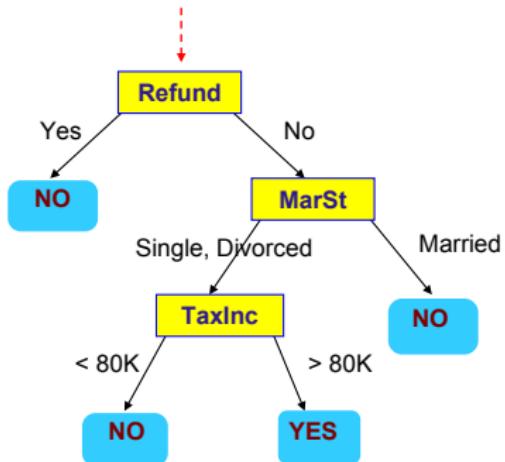
Tid	Refund	Marital Status	Taxable Income	Cheat	class
1	Yes	Single	125K	No	categorical
2	No	Married	100K	No	categorical
3	No	Single	70K	No	continuous
4	Yes	Married	120K	No	continuous
5	No	Divorced	95K	Yes	continuous
6	No	Married	60K	No	continuous
7	Yes	Divorced	220K	No	continuous
8	No	Single	85K	Yes	continuous
9	No	Married	75K	No	continuous
10	No	Single	90K	Yes	continuous



There could be more than one tree that fits the same data!

Apply Model to Test Data

Start from the root of tree.



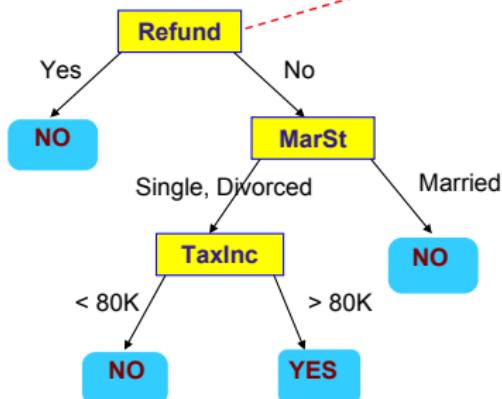
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

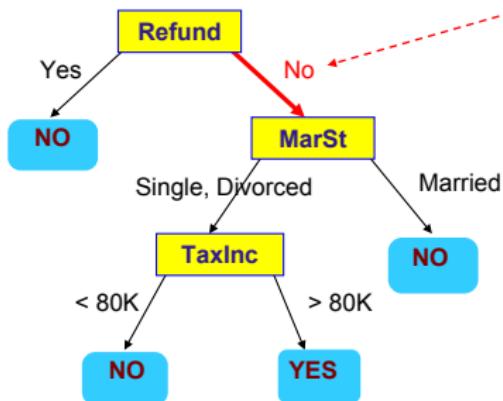
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

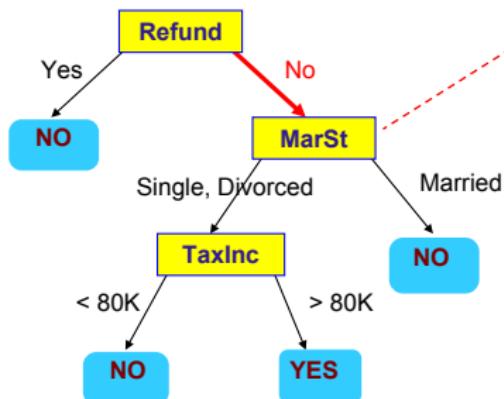
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

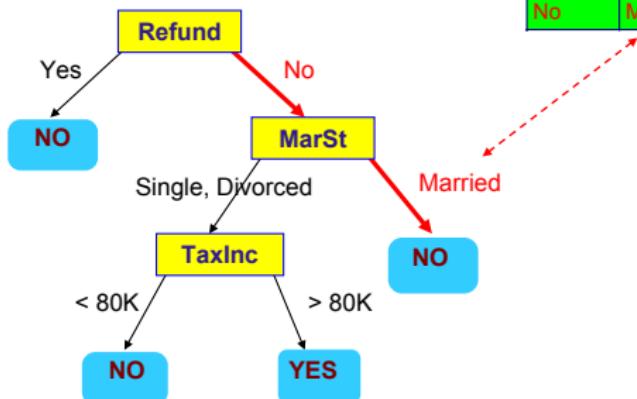
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

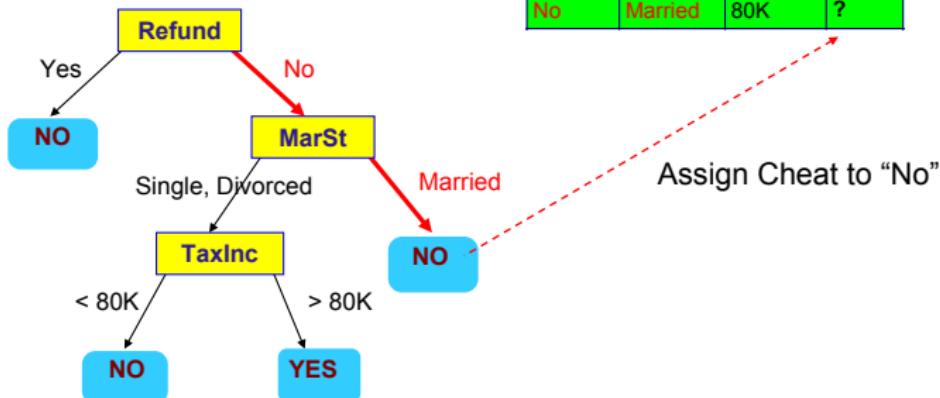
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Algoritmus döntési fa készítésére

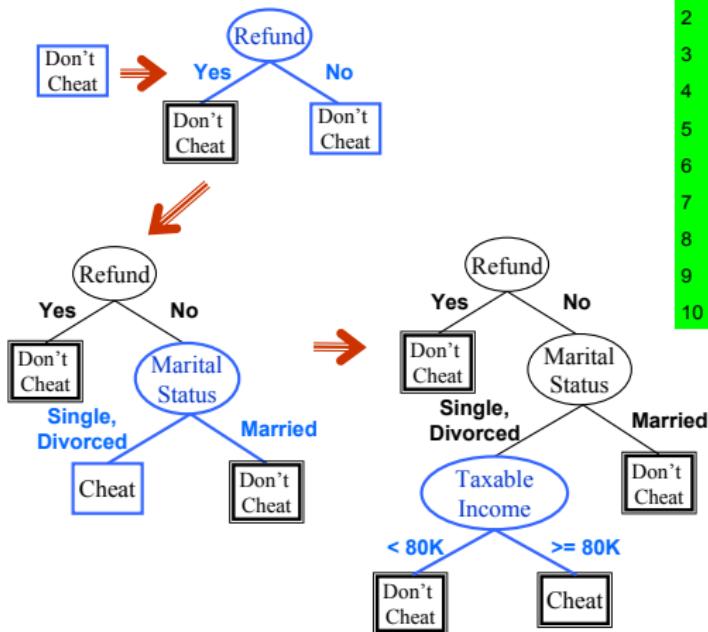
- egy általános algot nézünk, ennek különböző verziói futnak különböző programokban
- nem a legjobb fát akarjuk megtalálni (ez amúgy is aluldefiniált fogalom), hanem egy elég jót
- mohó módon, lokális döntéseket hozva, gyorsan

Hunt algoritmus, vázlat

- elején egy csúcs, ide tartozik minden rekord, címke a többségi címke
- később: választunk egy csúcsot, amit érdemes lenne szétvágni és valami attribútum mentén szétvágjuk egy vagy több részre
- vége: ha már nem érdemes vágni sehol

Hunt's Algorithm

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt algo, kérdések

- mikor érdemes vágni?
- melyik csúcsot vágjuk, ha több lehetőség is van?
- hogyan vágunk?
- az új csúcsokat hogyan címkézzük?
- mikor van vége?

Hunt algo, kérdések

- Mikor van vége?
 - Ha már nincs olyan csúcs, amit vágni érdemes.
- Mikor nem érdemes vágni?
 - Ha nem akarunk tovább osztani: minden rekord azonos cél-címkéjű
 - Ha nem tudunk tovább osztani: olyan sorok vannak különböző címkével, amiknek minden más attribútuma megegyezik
- Melyik csúcsot vágjuk, ha több lehetőség is van?
 - Valami bejárás szerint, pl. szélességi, mélységi.
- Hogyan vágunk?
 - Erről mindárt, ez érdekes.
- Az új csúcsokat hogyan címkézzük?
 - Többségi címkével.

Milyen attribútum mentén és hogyan vágjunk?

Fő elv: sokféle vágást kipróbálunk és a legjobb szerint vágunk.

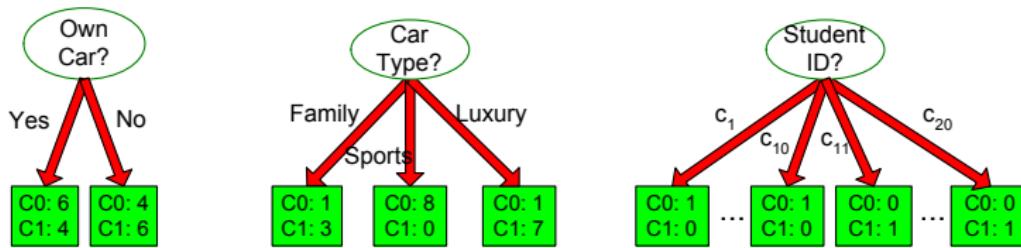
- Mik a lehetséges vágások? (Függ a szóba jövő attribútumok típusától.)
- Mi egy vágás jóságának a mértéke? Hogyan mérjük ezt?

Lehetséges vágások az attribútum fajtája szerint

- bináris attribútum: igen vagy nem, két gyerek csúcs lesz
- kategória típusú attribútum:
 - multiway split: minden lehetséges értékhez egy gyerek, az üres csúcsok címkéja a szülő többségi címkéje lesz
 - bináris split részhalmaz szerint: ebből van $2^t - 1$, ahol a t a lehetséges kimenetek száma
 - bináris vágás egy érték szerint: ebből van t darab (mint marital status az előző példában)
- folytonos attribútum
 - bináris vágás, az attribútum értéke kisebb-e egy adott küszöbnél (mint income az előző példában)
 - többes vágás: melyik sávba esik az érték

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

Vágás jósága, alapelvek

- többféle mérőszám van, mindegyik egy számértéket rendel a vágáshoz
- így a különböző vágások összehasonlíthatóak
- nagyrészt konzisztensek egymással a különféle mérőszámok
- mindegyik azt méri, hogy mennyire lesz homogén a létrejövő gyerek poluláció a célváltozó címkézése szerint

Vágás jósága

- 3 fő mérőszámot nézünk
- fő elv ugyanaz mindegyiknél:
 - egy rekordhalmazra definiálunk egy mérőszámot, ami az adott rekordhalmaz diverzitását mutatja (ebből lesz három féle mérőszám)
 - egy vágás jóságát azzal mérjük, hogy a szülőcsúcs diverzitása és a létrejövő gyerekcsúcsok diverzitása mennyire tér (mennyit nyerünk azon, ha szétvágjuk a szülőt egy adott módon, mennyivel lesznek homogénebbek a gyerekek)

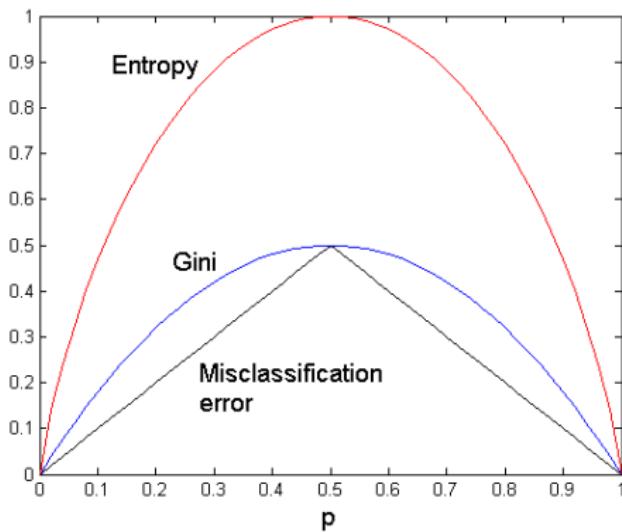
először azt nézzük, hogy hogyan lehet mérni egy rekordhalmaz, azaz a döntési fa egy csúcsának imhomogenitását

Inhomogenitás mérése: Gini-index, entrópia, classification error

- van egy t csúcsunk (egy rekordhalmaz), aminek egy c darab lehetséges értéket felvevő cél változó szerinti homogenitását akarjuk mérni
- p_i jelöli a rekordhalmazban előforduló i értékű rekordok relatív gyakoriságát
- $\text{Gini} = 1 - \sum_{i=1}^c p_i^2$
- $\text{entrópia} = - \sum_{i=1}^c p_i \log p_i$
- $\text{classification error} = 1 - \max_{i \in 1, \dots, c} p_i$

Comparison among Splitting Criteria

For a 2-class problem:



Vágás jósága

- ha már eldöntöttük, hogy melyik csúcsnál vágunk
- az adott csúcsnál minden lehetséges attribútum alapján, minden (?) lehetséges módon vágunk
- $\Delta = I(\text{szülő}) - \sum_{i=1}^k \frac{n_i}{n} I(\text{gyerek}_i)$
- itt $I()$ a három inhomogenitási mérték közül az egyik
- n_i az $i.$ gyerek rekordszáma, n a szülő rekordszáma
- arról van szó tehát, hogy a gyerekek inhomogenitását súlyozzuk a relatív nagyságukkal

ID alapján vágni?

- Azonosító alapján vágni (vagy más, nagyon kis elemszámú részhalmazra vágás) nem szerencsés:
 - ID esetén ez nem valódi nyereség
 - túl kicsi létrejövő halmazok esetén félő, hogy rosszul általánosít a modell (overfitting, erről mindenki beszél)
- de az algo ezt preferálja, mert itt lesz nagy a nyereség
- megoldások:
 - csak bináris vágás lehetséges
 - szűrjük ki a nyilvánvalóan felesleges attribútumokat (amik alapján nem akarunk úgyse vágni)
 - Δ_{info} helyett valami mással mérni a vágás jóságát: gain ratio

Gain ratio

- cél: büntessük azt, ha egy vágás túl sok részre vág szét
- ha entrópia az inhomogenitás mértéke: gain ratio
- gain ratio =
$$\frac{\Delta_{info}}{- \sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}}$$
- ez bünteti a szétszabdalást kicsi részekre

Az általános faépítő algo jellemzői

- gyorsan
- jól értelmezhető fát készít
- gyors az elkészült fával az osztályozás
- nem kell paramétereket beállítanunk előre (de)

Különböző programokban ennek verziói futnak:

- hogyan járjuk be a vágandó csúcsokat
- mi az inhomogenitás mértéke
- van-e multivágás vagy csak bináris

Leállási feltételek

- ha minden csúcs homogén
- ha nem tudunk tovább differenciálni
- valami globális leállási feltétel: levélszám, szintszámra korlát

A legjobb fa keresése nem kivitelezhető és nem is célszerű általában.

Overfitting

- nem szerencsés, ha a modell (pl. az elkészült döntési fa) túlságosan passzol a training set-re
- azért, mert nincs rá garancia, hogy a későbbi halmazok, amiken a modellt használjuk, teljesen ugyanilyenek lesznek (sőt...)
- egy egyszerűbb modell, aminek a training error-ja nagyobb, néha jobban általánosítható: jobban viselkedik a későbbi (a modell építése során nem látott) esetekre
- ez nem csak döntési fáknál léztező jelenség, hanem mindenhol, ahol egy training set alapján modellt építék, amit aztán korábban nem látott eseteken akarok használni

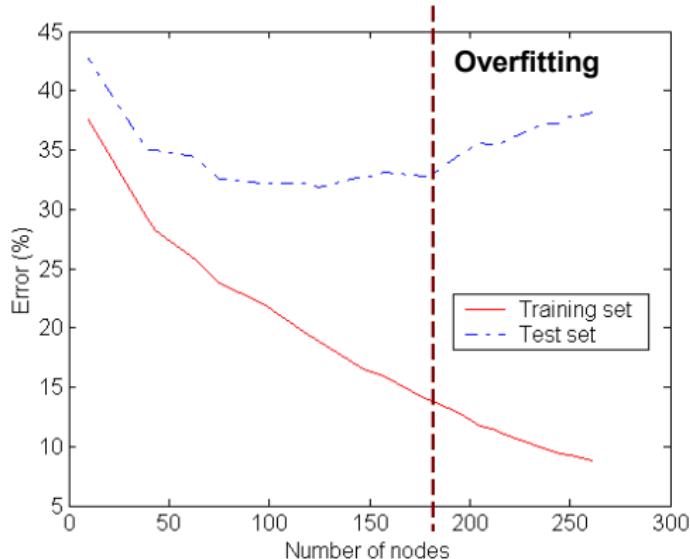
Training és test error

- felépítünk egy modellt (pl. egy döntési fát) aszerint, hogy mekkora a training error
- training error: a felépített modell hogyan osztályozza a training set rekordjait: accuracy, misclassification error
- de nekünk az lesz az érdekes, hogy a nem látott eseteken hogy viselkedik, hogy jelez előre
- ez lemérhető a teszt halmazon, de mi van, ha az derül ki, hogy ott nagy a hiba? (erről később)
- most az a fontos, hogy lássuk, hogy a training error és a test error két külön dolog

Underfitting és overfitting

- underfitting: a modell nem elég árnyalt ahhoz, hogy jól előrejelezzen
 - ekkor a training error és a test error is nagy
 - segíthet, ha bonyolultabb modellt építünk, ehhez esetleg kellhet több tanító adat
- overfitting: túlságosan jól passzol a modell a training set-re
 - csökkentsük a modell bonyolultságát valahogyan (erről mindenki beszél)

Underfitting and Overfitting

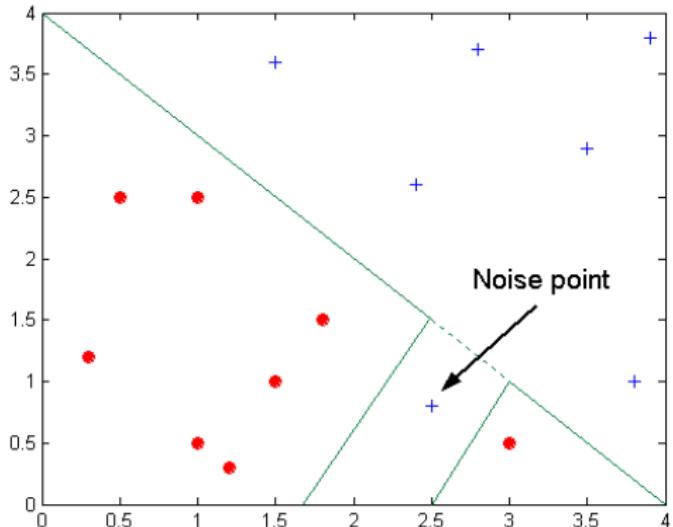


Underfitting: when model is too simple, both training and test errors are large

Overfitting oka

- közvetetten: túl erős, bonyolult modell, ami teljesen a training set-re szabható
- közvetlenül:
 - az adathalmaz nem reprezentatív (speciális esetek jelentős számban)
 - zaj
 - ha túl sok modell közül választhatunk: a legjobb nagyon jó lesz a training set-en, de semmi garancia nincs rá, hogy máshol is

Overfitting due to Noise



Decision boundary is distorted by noise point

Occam's razor

- mindenkorában jó lenne elkerülni a túl bonyolult modellt
- elv: Occam's razor (Occam borotvája): ha van két modell, amik kábé ugyanazt tudják (hol?, hogyan?) akkor az egyszerűbbet, kisebbet válasszuk
- a modell építése során vegyük ezt az elvet figyelembe, építsük a modellt úgy, hogy az büntesse a túl bonyolultat

Kérdés

- Hogyan vegyük figyelembe a modell teljesítményét a modell építése során?
 - Nehézség: eddig a modell értékelésére a teszt halmazt használtuk, de azt nem nézhetjük meg az építés alatt.
- Vezessünk be egy újfajta hibamérést, ami figyelembe veszi a modell bonyolultságát is a training error-on kívül: generalization error
 - resubstitution error (optimista becslés)
 - pessimista becslés (modellfüggő)
 - validation set

Resubstitution error

- feltételezzük, hogy a training set jól reprezentál
- optimistán azt gondoljuk, hogy az új adathalmazon is ugyanolyan jól fog osztályozni, mint a training set-en
- ekkor a modell hibája az, amit ennek addig hívtunk: misclassification error, azaz hány rekord lesz rosszul címkézve a training set rekordjai közül
- persze ha elég nagy fát (bonyolult modellt) építünk és kevés az adat, akkor ez simán lehet 0

Pesszimista verzió a generalization error-ra

- mivel a training set-re van szabva a modell, a valóságban nem lesz ilyen jó, nagyobb lesz a hiba
- az, hogy mennyire lesz rosszabb, az függ a modell bonyolultságától
- a pessimista hiba két tagból áll: a training errorból és egy másik tagból, ami a modell bonyolultságát bünteti
- döntési fáknál pl. lelevelenként egy plusz konstans taggal megnöveljük a hibásan osztályozott rekordok számát
- pl. lelevelenként 1: a training errorhoz hozzá kell adni $\frac{\ell \cdot 1}{n}$ -t, ahol ℓ a levelek száma, n pedig az összes rekord száma

Validation set

- a rendelkezésre álló adatokat nem két részre osztjuk (training és test), hanem háromra: training, validation és test
- ha két különböző modell (pl. két fa) között kell dönten, akkor a validation set-en megnézzük az előrejelzésüket és a jobbat választjuk
- azért kell erre külön halmaz (nem a test set), mert a test set-et a végső modell tesztelésére tartogatjuk

Hogyan veszem figyelembe a modell becsült generalized error-ját a modell építése során?

- pre-pruning: a fa építése közben nézzük, hogy a generalized error nő-e és ha igen, akkor nem vágunk
- post-pruning (teljes fát építünk, az általános algoval, amíg lehet, addig vágunk), aztán alulról felfele haladva visszavágjuk, ha a visszavágott fa hibája kisebb (pesszimista hiba vagy validation error)

Example of Post-Pruning

Class = Yes	20
Class = No	10
Error = 10/30	

Training Error (Before splitting) = 10/30

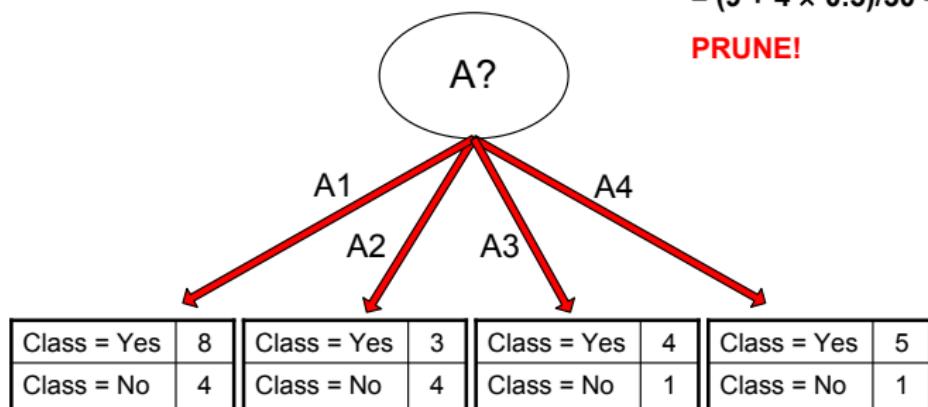
Pessimistic error = $(10 + 0.5)/30 = 10.5/30$

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

$$= (9 + 4 \times 0.5)/30 = 11/30$$

PRUNE!

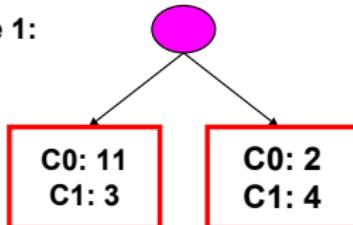


Examples of Post-pruning

- Optimistic error?

Don't prune for both cases

Case 1:



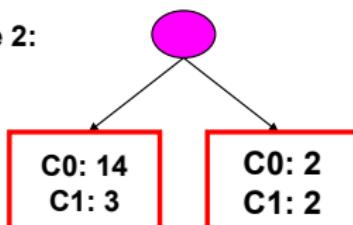
- Pessimistic error?

Don't prune case 1, prune case 2

- Reduced error pruning?

Depends on validation set

Case 2:



Misclassification error, változatok

- eddig accuracy és misclassification error volt: hibás előrejelzések száma az összes közül
- ez nem mindig jó, pl. ha nagyon kevés rekord van az egyik kategóriában és az az algo, hogy mindenkit a gyakori címkével címkézünk
- ezért költség-mátrixot is használhatunk, ha a hibás pozitív vagy hibás negatív osztályozást akarjuk nagyon büntetni
- vagy használhatunk accuracy helyett más (precision, recall, F-measure)

Cost Matrix

		PREDICTED CLASS		
		C(i j)	Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	C(Yes Yes)	C(No Yes)	
	Class>No	C(Yes No)	C(No No)	

$C(i|j)$: Cost of misclassifying class j example as class i

Computing Cost of Classification

Cost Matrix		PREDICTED CLASS		
		C(i j)	+	-
ACTUAL CLASS	+	-1	100	
	-	1	0	

Model M ₁	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%
Cost = 3910

Model M ₂	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%
Cost = 4255

Másik lehetőség az osztályozó jóságának mérésére

- true positive (tp): hány olyan rekord van, ami pozitív címkét kap és a valóságban is pozitív
- true negative (tn): hány olyan rekord van, ami negatív címkét kap és a valóságban is negatív
- false positive (fp): hány olyan rekord van, ami pozitív címkét kap, de a valóságban negatív
- false negative (fn): hány olyan rekord van, ami negatív címkét kap, de a valóságban pozitív

Ez a confusion matrix

Másik lehetőség az osztályozó jóságának mérésére

- accuracy ezzel a jelöléssel $\frac{tp + tn}{tp + tn + fp + fn}$
- precision (= p): hány eset valóban pozitív a pozitívnak mondottak közül, azaz $\frac{tp}{tp + fp}$
- recall (=r): hány pozitív esetet találunk meg tényleg: $\frac{tp}{tp + fn}$
- F-measure = $\frac{2rp}{r + p} = \frac{2tp}{2tp + fp + fn}$

R-ben mi van?

- sok minden :)
- több package döntési fa készítésre: tree, rpart, party
- alap-parancsal létrehozok egy fa típusú objektumot: megadom, hogy milyen training set-en, milyen változókat vegyen figyelembe
- a létrehozott fát használhatom előrejelzésre, ábrázolhatom
- lehet egyszerűsíteni (prune)

tree package

```
> library(tree)

> ir.tr <- tree(Species ~., iris)
> summary(ir.tr)
Classification tree:
tree(formula = Species ~ ., data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width" "Sepal.Length"
Number of terminal nodes:  6
Misclassification error rate:  0.02667 = 4 / 150

> plot(ir.tr)
> text(ir.tree)
```

kNN osztályozás

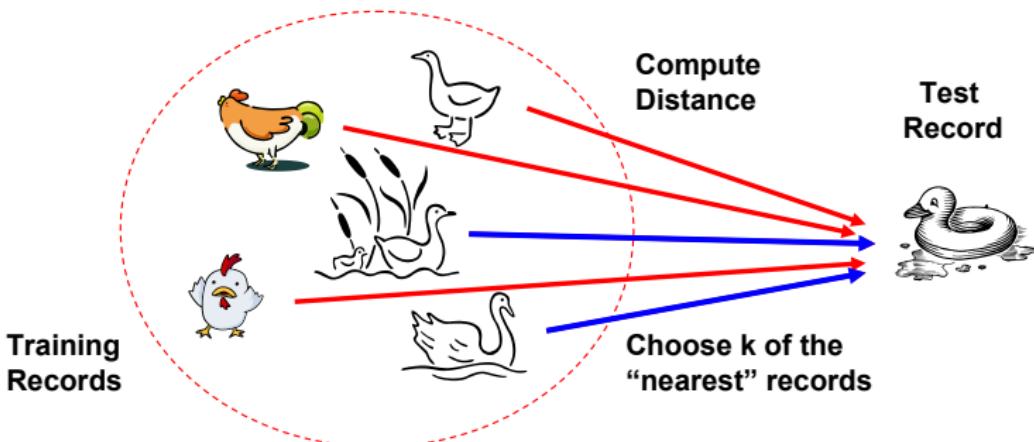
Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2018. március 12.

Nearest Neighbor Classifiers

- Basic idea:
 - If it walks like a duck, quacks like a duck, then it's probably a duck



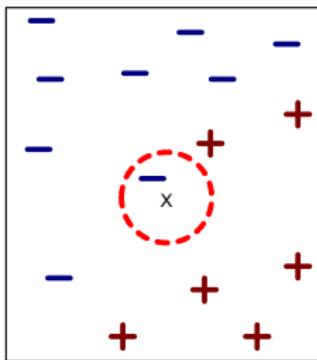
Elve

- a rekordok pontok az annyi dimenziós térben, ahány attribútum van (az osztályt nem számítva)
- az osztályozandó sor címkéje a hozzá legközelebb eső k darab training record alapján lesz valahogyan

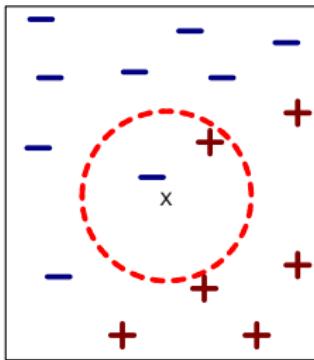
Mi kell ehhez?

- az összes training record
- milyen távolságot használunk?
- mi legyen a k ?

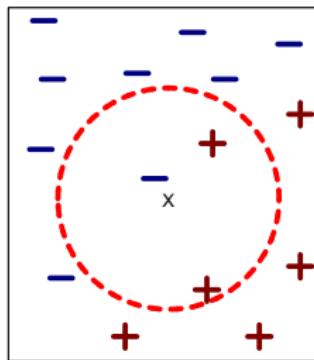
Definition of Nearest Neighbor



(a) 1-nearest neighbor



(b) 2-nearest neighbor



(c) 3-nearest neighbor

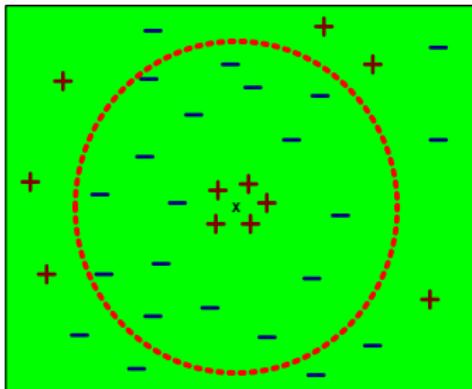
K-nearest neighbors of a record x are data points that have the k smallest distance to x

Kérdések

- távolság:
 - euklideszi (skálázás kellhet)
 - SMC, Jaccard vagy más, amit tanultunk: az a lényeg, hogy azok legyenek közeliek, akiket annak gondolunk
- döntés a címkéről
 - többségi szavazás a k szomszéd között
 - súlyozott szavazatok: $w_i = \frac{1}{d_i^2}$, ahol a d_i az $i.$ szomszéd távolsága
- k
 - kicsi k esetén érzékeny a lokális hibákra
 - nagy k esetén bezavarhat sok távoli

Nearest Neighbor Classification...

- Choosing the value of k:
 - If k is too small, sensitive to noise points
 - If k is too large, neighborhood may include points from other classes

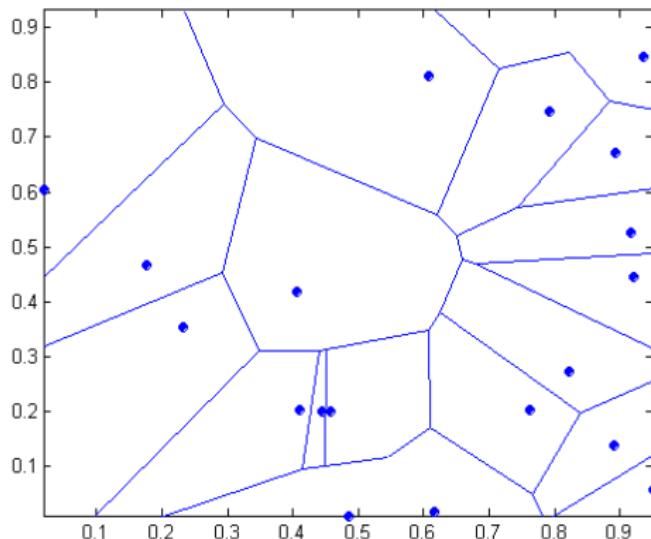


Összefoglalás

- lazy learner: csak akkor dolgozik, amikor osztályozandó sor jön
- lassú (drága) egy sor osztályozása, de nincs hosszú előkészítés
- lehet előkészítéssel gyorsan osztályozni, ha $k = 1$: felosztjuk a teret cellákra, minden térbeli ponthoz hozzárendeljük a hozzá legközelebbi training recordot: Voronoi-diagramm

1 nearest-neighbor

Voronoi Diagram



Naive Bayes

Feltételes valószínűségek szorzási szabálya

Feltételes valószínűség: $P(A|B) = P(A,B)/P(B)$

3 változóra:

$$P(A,B,C) = P(A|B,C) * P(B|C) * P(C)$$

N változóra:

$$P(x_1, x_2, \dots, x_N) =$$

$$P(x_1|x_2, x_3, \dots, x_N) * P(x_2|x_3, \dots, x_N) * \dots * P(x_{[N-1]}|x_N) * P(x_N)$$

Bayes téTEL

Bayes téTEL: $P(A|B) = P(B|A)P(A) / P(B)$

Naive Bayes

$x_1, x_2, \dots, x_N (=x)$, y - valószínűségi változók, az adatunk

$P(y|x)$ -et keressük, és a predikciónk $\operatorname{argmax}\{y\} P(y|x)$ (Bayes téTEL)

$$P(y|x) = P(x|y)P(y) / P(x) \quad (\text{itt } P(x) \text{ konstans})$$

$$P(x|y)P(y) = P(x_1, x_2, \dots, x_N, y) = P(x_1|x_2, \dots, x_N, y)P(x_2|..., y) * \dots * P(y)$$

Naive: feltételezzük x -ek kölcsönös függetlenségét:

$$P(x|y)P(y) = P(x_1|y) * P(x_2|y) * \dots * P(x_N|y) * P(y)$$

Tehát:

$P(y|x)$ arányos $P(x_1|y) * P(x_2|y) * \dots * P(x_N|y) * P(y)$ kifejezéssel

$P(x|y)$ és $P(y)$ közelítése (Gaussian Naive Bayes)

$P(x|y)$ -hez Maximum Likelihood becslést használunk:

Pl. x folytonos, közelítsük $P(x|y)$ -t $N(\mu_y, \sigma_y)$ eloszlással,

ML feladat: $\operatorname{argmax}\{\mu, \sigma\} P(x|y, \mu, \sigma)$

Megoldás: $\mu_y = \operatorname{mean}(x|y)$, $\sigma_y = \operatorname{std}(x|y)$

$P(y)$: bármilyen prior választhatunk (pl. domain knowledge)

Numerikus stabilitás miatt log valószínűségekkel dolgozunk

A logaritmus függvény szigorúan monoton,
ezért $\operatorname{argmax}\{y\} P(y|x) = \operatorname{argmax}\{y\} \log(P(y|x))$

A logaritmus szorzási azonosságot alkalmazhatjuk:

$$\log(P(x_1|y)*P(x_2|y)*...*P(x_N|y)*P(y)) = \log(P(x_1|y)) + \log(P(x_2|y)) + \dots + \log(P(y))$$