Data preprocessing and data analysis methods in Big data interfaces

Nataliya Shakhovska

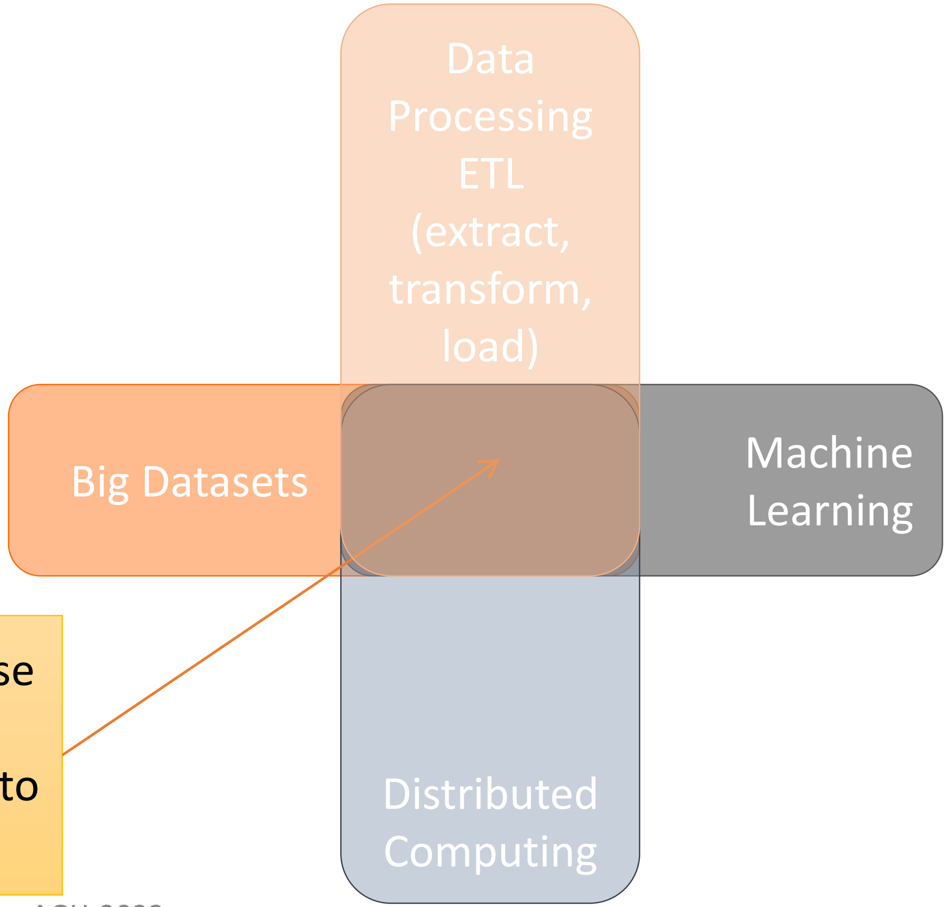
Agenda

- Intro
- Integration
- Missing data imputation method
- Feature selector
- Predictive ensemble
- Conclusion

Data Processing and Machine learning Methods

- Data processing (third trend)
  - Traditional ETL (extract, transform, load)
  - Data Stores (HBase, ……..)
  - Tools for processing of streaming, multimedia & batch data
- Machine Learning (fourth trend)
  - Classification
  - Regression
  - Clustering
  - Collaborative filtering

Working at the Intersection of these four trends is very exciting and challenging and require new ways to store and process **Big Data**



?

By the way, what is about nature of Big data?

What is heterogeneous data?

Heterogeneous Data is data from any number of sources, largely unknown and unlimited, and in many varying formats. In essence, it is a way to refer to data that id of an unknown format and/or content.

Structured data

MDX
PLSQL
SQLJ
SQL

Database
Datawarehouse

noSQL

Semi-structured data

SPARQL
RQL
SeRQL
TRIPLE
RDQL
N3
Versa

XML
HTML
Spreadsheets (xls)
RDF

Unstructured data

Parser

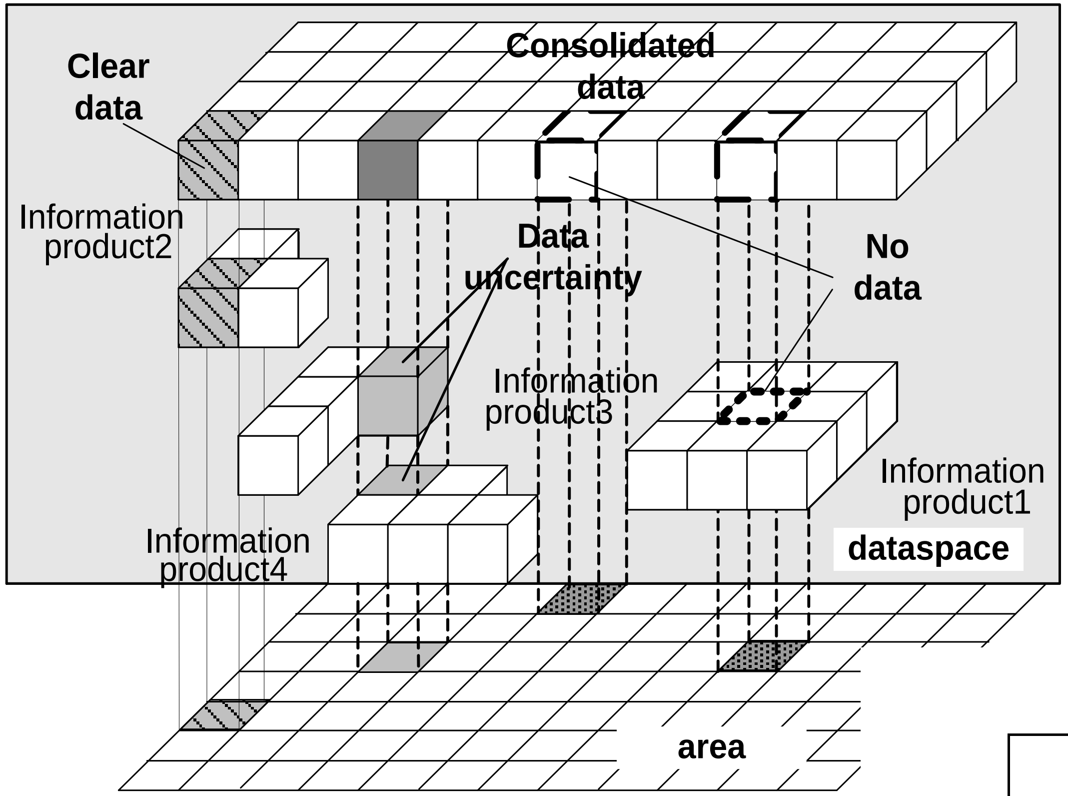
Find by keywords

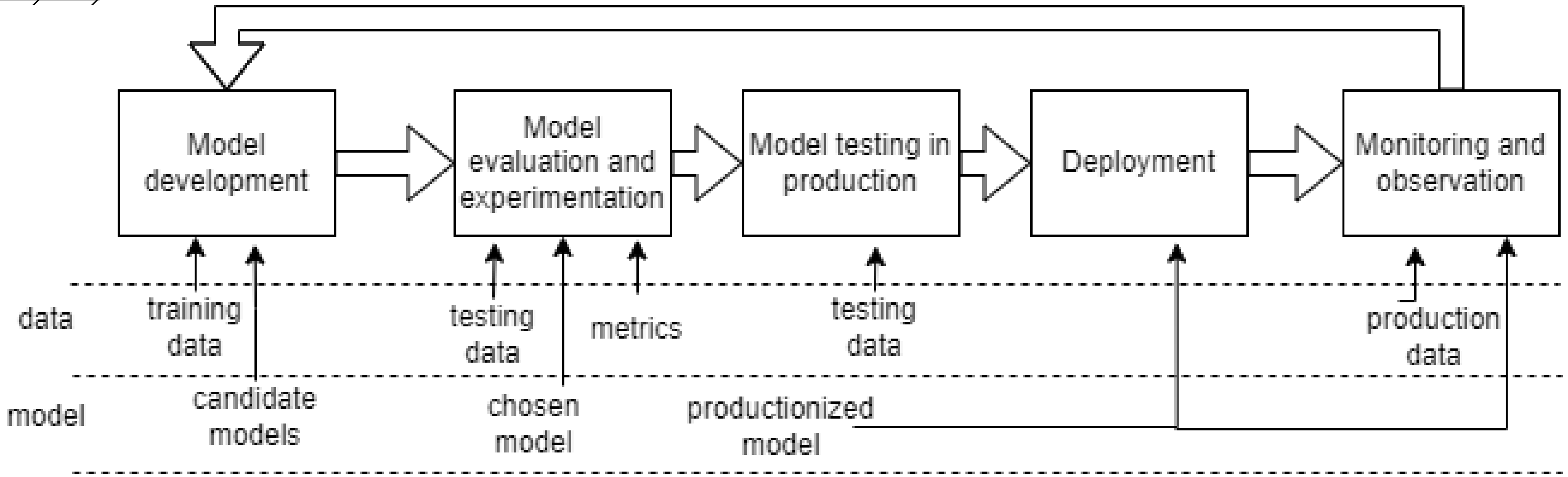
Text
Video
Special format

Different issues:
- Relation algebra
- Spatial operations
- Search in array
- Search in tree/graph
- Propagation

Types of dependency:
- Independent - time
- Serial - spatial

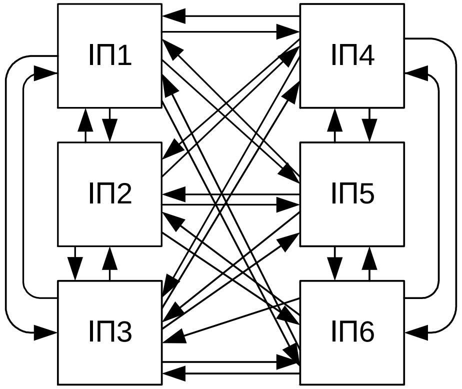
Big data VS consolidation



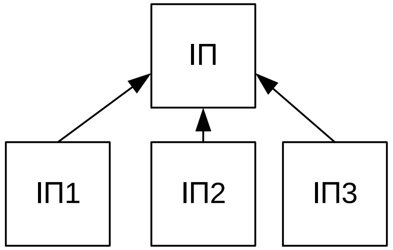


Problems with data gathering

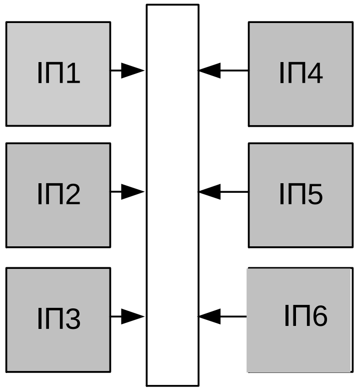
- Description of the information object and its characteristics
  metadata, dictionary of synonyms
- Retrieving information from the object
  methods for data transforming
- Object that can exchange information
  data protocols
- Quality data object in the context of a complex information system
- Ability to change the organization of objects



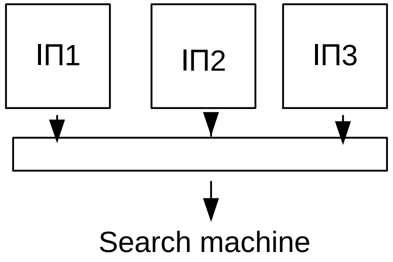
$C = c_n, T = t_n, Q(q_1 ,\ldots, q_n) \to \max$



$C = c_i, T = t_i, q_i \to \max, q_j \to \min$



$C = c_i, T = t_i, Q(q_1,\ldots,q_n) \to \max$



The proposed steps of consolidation (structured and semi-structured data)

- determine the type of source
  classification
- definition of data structures
  XML
- comparison of data structures

relation algebra

Intelligence agent

$$f_{Ip}(DS) \xrightarrow{Agent} Cg \cup Ip.Cg$$

Agent = < ***Cg***, EM, *Dic*, *Experience_Base*,*Solver*, *Effector* >

Experience_Base = $\sigma_{evdate=Date()}(Dic)$

- propagate the updates from the data sources to the warehouse
- Confidence evaluation

Trust theory

Intelligence agent

$$Trust_j = \frac{\sum_{i=1}^{m} \left(\frac{\sum_{k=1}^{n} Trust_k(i, j)}{n} \right)}{n * m}$$

Existing methods of data imputation

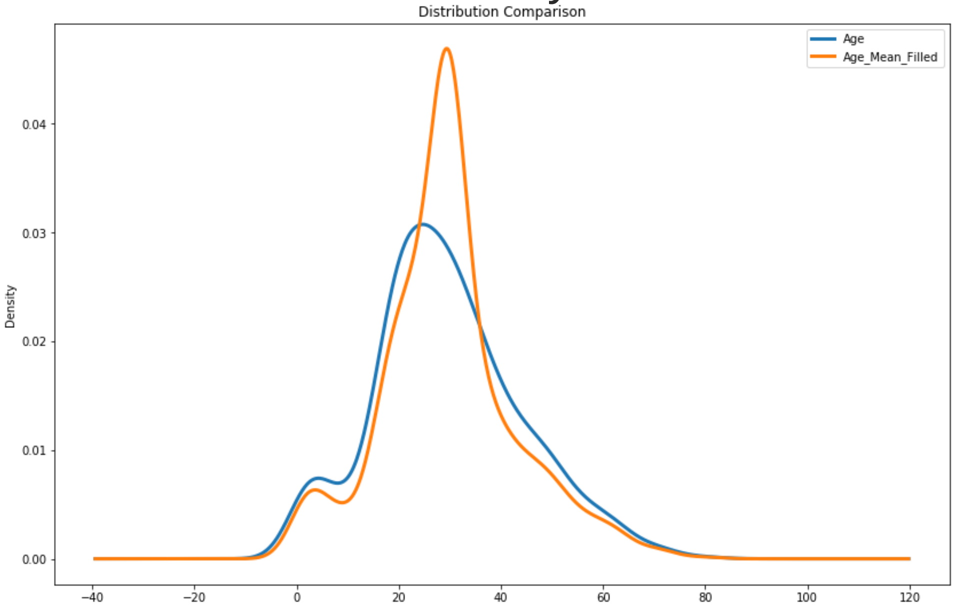
```
count     714.000000
mean       29.699118
std        14.526497
min         0.420000
25%        20.125000
50%        28.000000
75%        38.000000
max        80.000000
Name: Age, dtype: float64
```

Mean Imputation

```
count     891.000000
mean       29.699118
std        13.002015
min         0.420000
25%        22.000000
50%        29.699118
75%        35.000000
max        80.000000
Name: Age_Mean_Filled, dtype: float64
```

- *Mean substitution*
- *Hot-deck imputation* (randomly)
- *Cold-deck imputation* (constant)
- Model-based
  - Regression
  - EM
  - RF
  - KNN
  - SOM
  - SVM

mean imputation has an impact on attributes variability



The Big Data Model in the Task of Missing Data Recovery

The Big data schema Bd is the finite set of attributes with exact values $\{A_1, A_2, ..., A_n\}$, set of attributes with indefinite, inexact or missing values $\{A\_unk_1, A\_unk_2, A\_unk_p\}$ and the set of values of membership functions for inexact attributes $\{Unk_1, Unk_2, \ldots, Unk_m\}$. In addition, the catalog of the attributes (features) with schema Cg and synonym dictionary with schema Dic should be used:

$$Bd = < \{A_1, A_2, ..., A_n\}, \{A\_unk_1, A\_unk_2, A\_unk_p\}, \\ , \{Unk_1, Unk_2, \ldots, Unk_m\}, Dic, Cg > \quad (1)$$

The attributes of the $\boldsymbol{A\_unk}$ set are considered indeterminate, and the level of confidence in them is stored in the attributes of the set $\boldsymbol{Unk}$.

A binary relation Rel is used to show the relationships between the attributes of the $\boldsymbol{A\_unk}$ and $\boldsymbol{Unk}$ sets, the values of which are determined based on the sample view of the source and in the Cg data directory:

$$Rel = \left| rel_{ij} \cdot \sigma_{\arg(i)}(Cg) \right|, \forall i = \overline{1,p}, \forall j = \overline{1,m} \\ rel_{ij} = \begin{cases} 1, \ Unk_j \Leftrightarrow A_{unk_i} \wedge \sigma_{\arg(j)}(Dic) \\ 0, \ otherwise \end{cases} \quad (2)$$

Examples of consolidated data tuples for different types of information resources

1. Relational database - in this case an extended relational tuple is used t_{rel} :

$$bd = t_{rel} \cup Unk,$$
$$t_{rel} = \{a_1, ..., a_n\} \cup \{a\_unk_1, ..., a\_unk_m\}, \quad (4)$$

where $\{a_1, \ldots, a_n\}$ are the value of exact attributes, $\{a\_unk_1, \ldots, a\_unk_m\}$ are the value of attributes with uncertainty.

2. Data Warehouse combines fact and dimension data. A set of measurement values and fact characteristics is presented as a tuple t_{dw} :

$$bd = t_{dw} \cup Unk,$$
$$t_{dw} = \{a_1, ..., a_n\} \cup \{a\_unk_1, ..., a\_unk_m\} \cup$$
$$\cup \{a_{rf1}, ..., a_{rfi}\} \cup \{a\_unk_{11}, ..., a\_unk_{1m}\} \cup ... \quad (5)$$
$$... \cup \{a\_unk_{k1}, ..., a\_unk_{ks}\} \cup ...$$
$$... \cup \{a\_unk_{rf1}, ..., a\_unk_{rft}\} \cup,$$

where $a_{i,j}$ is the value of exact characteristic j in the dimension i , a_{rf1} is the value of the characteristic j of fact table, $a\_unk_{i,j}$ is the value of attribute with uncertainty j from dimension i , a_unk_{rfj} is value of attribute with uncertainty j from the fact table.

3. Semi-structured text describes the values of the nodes of the semantic networks and the degree of affiliation of these values to the objects whose names are described in the synonym dictionary t_{text} :

$$bd = t_{text} \cup Unk,$$
$$t_{text} = \{a_1, ..., a_n\} \cup \{a_{unk_1}, ..., a_{unk_m}\}, \quad (6)$$

Probabilistic Production Dependencies Mining

Probabilistic Production Dependency is a production rule in the basic ratio selection that is valid for a significant number of entities in that selection. The significance threshold should be determined expertly, or based on calculations of the probability of erroneous selection of this dependence. The main difference between associative rules and PPD is that PPD will generated from existing FD in dataset.

$$\begin{aligned} &F_I : K = \{a_i\}, a_i \in A, D = \{a_j\}, \\ &a_j \in A, : P(k \in K \to d \in D) = p \end{aligned} , \tag{8}$$

where k and d are the tuples of groups of attributes K and D, respectively.

The main indicator of the reliability of such a dependency is the ratio of the number of objects that such a PPD has to the number of objects in the selection:

$$P(F_I) = \frac{\left|\sigma_{k \in K \wedge d \in D}(R)\right|}{\left|\sigma_{k \in K}(R)\right|}. \tag{9}$$

The classification rule is called the probabilistic productive relationship between the subsets of the *X* and *Y* attributes in the consolidated Big data *Bd*, which occurs in training set *bd* with trust level *s*, where:

$$(X = x) \to (Y = y). \tag{10}$$

Association rules measures

Trust Level is the ratio of the number of objects that have such a PPD to the number of objects in the selection:

$$Conf\left(S \to T\right) = P\left(S \to T\right) = \frac{\left|\sigma_{S \wedge T}\left(r\right)\right|}{\left|\sigma_{S}\left(r\right)\right|}. \tag{11}$$

Support Level is the characteristic of a selection predicate in a ratio that is calculated as the ratio of the number of objects that satisfy P predicate to the total number of objects in relation to:

$$Supp\left(P\right) = \frac{\left|\sigma_{P}\left(r\right)\right|}{\left|r\right|}. \tag{12}$$

When calculating the level of support for PPD, the conditional and the resulting dependency predicate are combined by a conjunction:

$$Supp\left(S \to T\right) = Supp\left(S \wedge T\right) = \frac{\left|\sigma_{S \wedge T}\left(r\right)\right|}{\left|r\right|}. \tag{13}$$

Using this concept, the *level of trust* can be calculated as:

$$Conf\left(S \to T\right) = \frac{Supp\left(S \to T\right)}{Supp\left(S\right)}. \tag{14}$$

The *level of improvement* is calculated as the ratio of the levels of trust and support of the PPD:

$$\mathrm{Imp}\left(S \to T\right) = \frac{Conf\left(S \to T\right)}{Supp\left(T\right)} = \frac{Supp\left(S \wedge T\right)}{Supp\left(S\right) \cdot Supp\left(T\right)}. \tag{15}$$

Total *mutual information* is generally defined as:

$$I_{X \leftrightarrow Y} = \sum_{i=1}^{n}\sum_{j=1}^{m} P_{ij} \log_2 \frac{P_{ij}}{p_i r_j}, \tag{16}$$

Algorithm 1: PPD mining

Input: Big data dataset *bd*, Card(*Bd*)=*n*

Output: *PPDlist*

1:*PPDlist*={}

2:X={}

3:**for** (i=1;i<n;i++)

4: $X = X \cup A_i$

5: Group entities with the same values for the set of attributes *X*;

6: Search for entities that have the same values for the set of synonyms of attributes *X*;

7: *Y*={}

8: **for** (j=i+1;j<=n;j++)

9: $Y = Y \cup A_i$

10: Calculating *Supp* and *Conf* in the source of the entities obtained in steps 5) and 6);

11: Calculating the *Imp* of the tuple sources obtained in steps 5) and 6);

12: Identifying the entities with the highest levels of confidence and add X→Y to *PPDlist*.

Algorithm 2: Recovery algorithm

Input: Big data dataset *bd* with schema *Bd and missed data* ⊥, PPDList

Output: Completeness level of *bd*

1:Completeness=0

2:**If** $(bd)=\{bd_1(X_1)\downarrow,\ ...,\ bd_1(X_n)\downarrow\}$ and $\{bd_2(X_1)\downarrow,\ ...,\ bd_2(X_n)\downarrow\}$ and $\{bd_1(X_1)\downarrow,\ ...,\ bd_1(X_n)\downarrow = bd_2(X_1)\downarrow,\ bd_2(X_n)\downarrow\}$ and $bd_1(Y)\downarrow$ and $bd_2(Y)=\perp\}$ and $\sigma_{X_1}(Dic)=\varnothing$ and $\{X_1 \to Y\ in\ \ PPDlist\}$

3:**then**

change ⊥ to $bd_1(Y)$

$$bd_1(P)=bd_1(P)/\left(\sum_i \frac{m_{1i}}{n}\right)$$

Completeness++

4:**If** $\{bd_1(X_1)\downarrow,\ ...,\ bd_1(X_n)\downarrow\}$ **and** {*m* from *n* values of attributes are ↓ in *bd*$_2$, *n* - *m* values of attributes are ⊥, $m \le n$} **and** $\left\{P \ge 1-{}^{m}\!/_{n}\right\}$ **and** {using defined values $bd_1(X^m)\downarrow,\ ...,\ bd_2(X^m)\downarrow\}$ } **and** $bd_1(Y)\downarrow,\ ...,\ bd_2(Y)=1\}$ **and** {X_2 →*Y* in *PPDlist*}

5:**then**

change ⊥ to $bd_2(Y)$

$$bd_2(P)=bd_2(P)/\left(\sum_i \frac{m_{2i}}{n}\right)$$

6: **If** { m_i from *n* values of attributes are ↓ in bd_i, $m_i \le n$ } **and** { m_j from *n* values of attributes are ↓ in bd_j, $m_j \le n$ }

and {for exact values $bd_i(X^m)\downarrow = bd_2(X^m)\downarrow\}$ **and** {for exact values $bd_j(X^m)\downarrow = bd_2(X^m)\downarrow$} **and** $\left\{{}^{m_i}\!/_{n} \le {}^{m_j}\!/_{n}\right\}$ **and** $\left\{P \ge 1-{}^{m_i}\!/_{n}\right\}$,

and $\{bd_i(Y)\downarrow\}$ and $\{bd_j(Y)\downarrow\}$ and $\{bd_2(Y)=\perp\}$, **and** $\{X_2, X_j \to Y\ \ in\ \ PPDlist\}$

7:**then**

change ⊥ to $bd_j(Y)$

Completeness++

Sequential dependencies

We determine the *parent moving* operator

$$Up_{c_{X=x_1,\sigma_X(Dic)}(bd)} = \sigma_{X=x_1}(\sigma_{X=Y,\sigma_X(Dic)}(bd)), \qquad (27)$$

where χ_1 is the value of primary key (child); χ_2 is the foreign key (parent), and

the *child moving* operator

$$Down\_c_{X=x_2,\sigma_X(Dic)}(bd) = \sigma_{Y=x_2,\sigma_X(Dic)}(bd). \qquad (28)$$

Based on the above described, we may build the operator for recovering the missing data in the sequence of the events (or entities):

$$\sigma_{X=x_1,Val=v,\sigma_X(Dic)}(bd) = \\ = Heir\_c\begin{pmatrix} \sigma^{cons}_{X=x_1,Val=v,\sigma_X(Dic)}(bd), \\ Down\_c_{X=x_1,Val=Null,\sigma_X(Dic)}(bd) \\ Bd.Unk_X = \\ = Recovery\left(Bd.Unk_X, P^X\left(\sigma_X\left(Dic\right)\right)\right) \end{pmatrix} \qquad (29)$$

Complexity estimation

$$t = O\left(t_{stat} + t_{ma}\right)$$

At this stage, the input relation with the data is passing through each tuple, so the time of this stage is directly proportional to the relation size *n*.

$$t_{stat} = O\left(n \cdot m^{H_{stat}}\right) = O\left(n \cdot m^{\log_{avgD}\left(\frac{n}{minSupport}\right)}\right) =$$

$$= O\left(n \cdot \left(\frac{n}{minSupport}\right)^{\log_{avgD}(m)}\right) =$$

$$= O\left(minSupport \cdot \left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)}\right).$$

$$M_{stat} = O\left(\left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)}\right)$$

If we use the Functional Dependencies as Elemental Dependencies for PPD Generation Enables Complexity, then:

$$t_{ma} = O\left(\frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{m \cdot D(A)}\right)$$

$$M_{ma} = O\left(Z_{ma} \cdot sz_{ma}\right)$$

$$\Rightarrow$$

$$t = O\left(\begin{array}{l} minSupport \cdot \left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + \\ + \frac{Z_{el}^2}{m \cdot D(A)} + \frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{m \cdot D(A)} \end{array}\right) =$$

$$= O\left(\begin{array}{l} minSupport \cdot \left(\frac{n}{minSupport}\right)^{1+\log_{avgD}(m)} + \\ + \frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{m \cdot D(A)} \end{array}\right).$$

Parallel mode – time complexity estimation

$$t_k = O\left(\begin{array}{l} minSupport \cdot \left(\dfrac{n}{k \cdot minSupport} \right)^{1+\log_{avgD}(m)} + \\ + \log_2(k) \cdot \left(\dfrac{n}{k \cdot minSupport} \right)^{1+\log_{avgD}(m)} \end{array} \right) +$$

$$+ O\left(\frac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{k \cdot m \cdot D(A)} \right) =$$

$$= O\left(\begin{array}{l} \left(minSupport + \log_2(k)\right) \cdot \left(\dfrac{n}{k \cdot minSupport} \right)^{1+\log_{avgD}(m)} + \\ + \dfrac{Z_{aggr}^2 \cdot \log\left(sz_{aggr}\right)}{k \cdot m \cdot D(A)} \end{array} \right)$$

In the case of parallel computing on a large number of processors, we can neglect by the second member of the function t_n. For the same reason, $log_2(n) = O(minSupport)$. The asymptotic estimate of the algorithm execution time (on a system of *k* processors)

$$t_n = O\left(minSupport^{-\log_{avgD}(m)} \right)$$

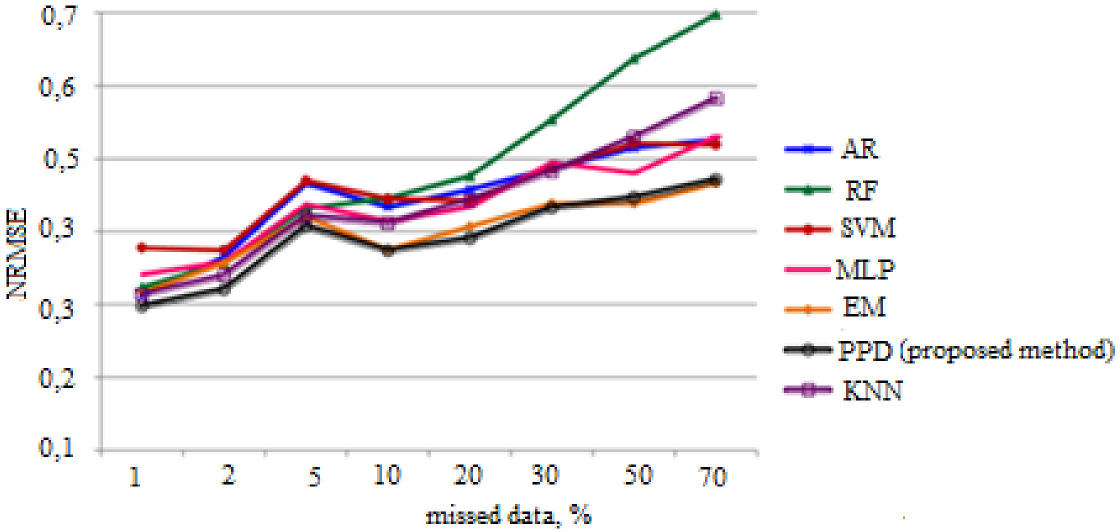
Experimental Results and Discussion

To run the experiment, the dataset from the Big Cities Health Inventory Data Platform ("BCHC Data Platform") is used. The platform contains over 18,000 data points across more than 50 health, socio-economic, urban (information from smart sensors) and demographic indicators across 11 categories in the United States. This is an example of semictructured data with hidden dependencies inside entities and between entities.

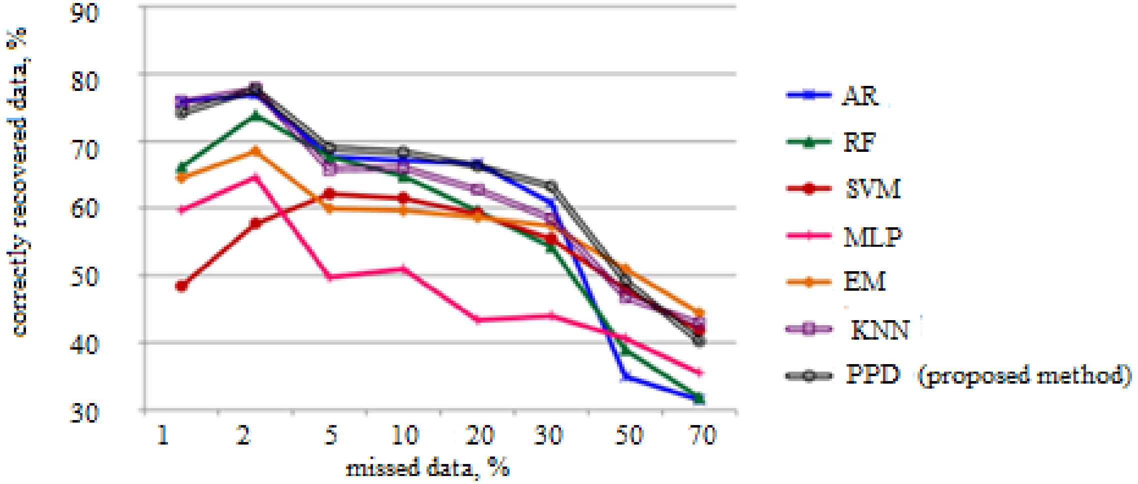
The missing data are modeled as random deleting of exact data.

The proposed and existing methods are tested on the same hardware: Intel Core 5 Quad E6600 2.4 GHz, 16 GB RAM, HDD WD 2 TB 7200 RPM. The criteria of PPD creation are $Conf(F_1)$>0.7 and *minSupport*=100. RStudio is used for data modelling and analysis.

Comparison with existing approach



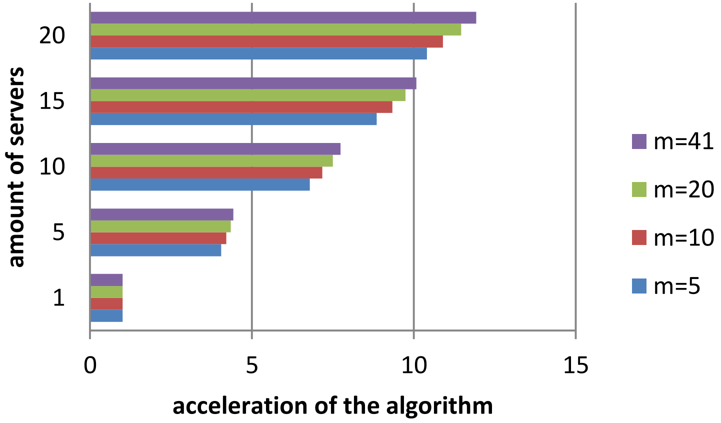
normalized root-mean-square error



The analysis of the percentage of correctly recovered data

Amount of records	SVM	AR	EM	PPD (proposed method)
2 000	31	190	9	8
4 000	49	362	23	19
6 000	95	437	37	31
8 000	144	639	49	42
10 000	210	827	62	51
18 000	286	1019	77	65

Time of analysis (*min*), depending on the amount of analyzed data

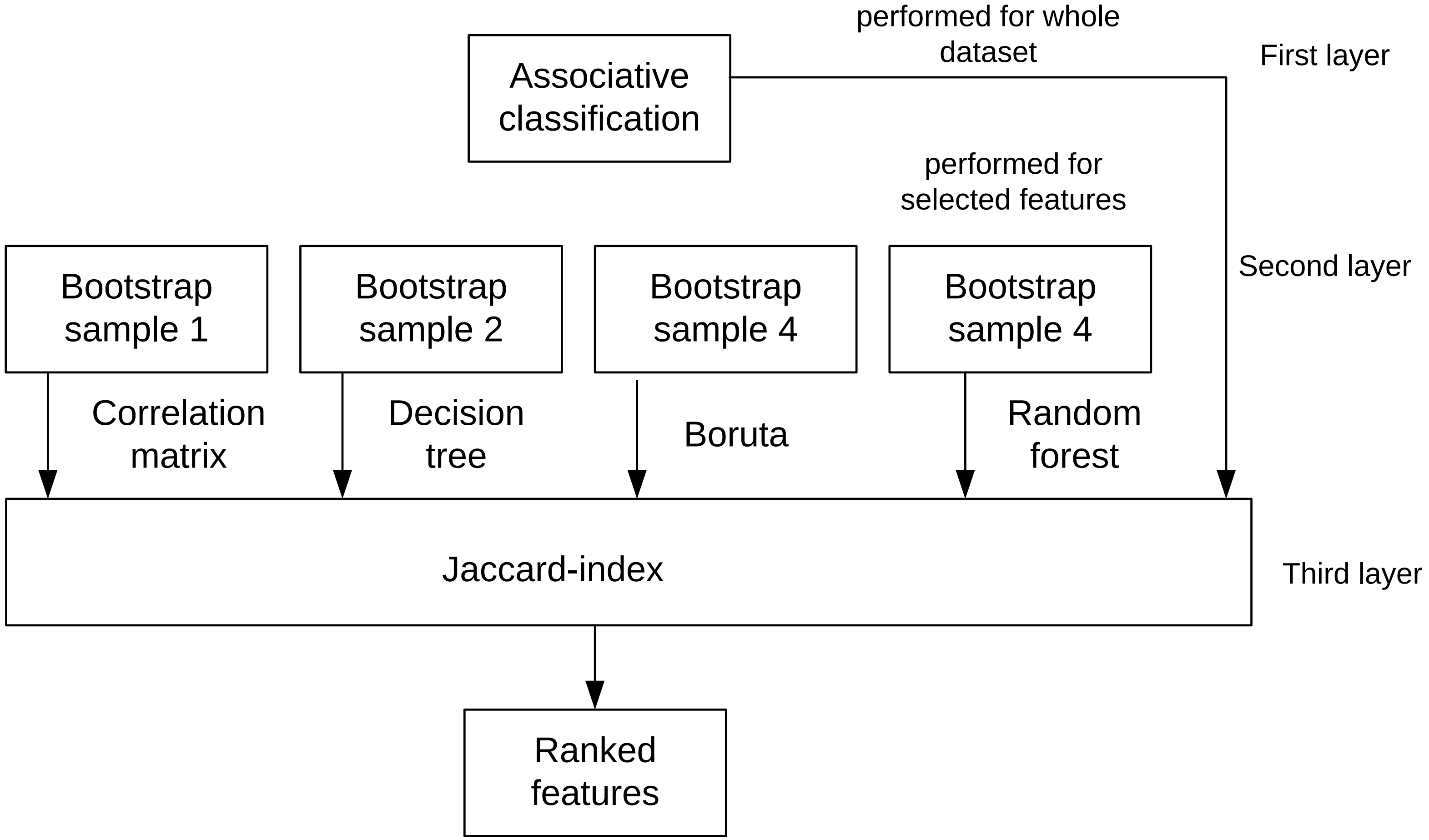


The acceleration graph of the developed algorithm

Feature selection: overview

- filters,
- wrappers,
- built-in algorithms

The hybrid ensemble feature selection model



$$(S_1, \ldots, S_{1n}) = \frac{|S_1 \cap \cdots \cap S_n|}{|S_1 \cup \cdots \cup S_n|}$$

Problem statement

- Dataset consists of 35 features and 122 instances collected from Lviv regional rehabilitation center for post-COVID patients with short- and long-term (more than 20 days) treatment and rehabilitation.
- The personal data were removed from the dataset and replaced with unique random identifiers.
- The next feature, sex, is processed using one-hot encoding technics and in the final dataset is presented in two components – female and male.
- Features like age, weight, height, BMI, CAT, pulse, the function of external respiration are taken as physiological parameters measured before inpatient treatment

TNF-α, pg/ml	6.15±1.20
IL-8, pg/ml	15.70±2.00
IL-4, pg/ml	16.10±1.13
IL-10, pg/ml	36.60±1.96
TNF-α+IL-8+IL-4+IL-10	0.65±0.04
CD3+, %	66.20±0.60
CD22+, %	15.20±0.29
0-lymphocytes, %	18.70±0.65
CD4+, %	38.10±0.67
CD8+, %	27.20±0.39
CD4+/CD8+	1.410±0.036
CD3+/CD22+	4.39±0.11
(CD3++CD22+) 0-lymphocytes	4.48±0.22
CD16+, %	17.10±0.44

Associative rules mining

№	Rule	Supp
1	{CD4=[26,28)}	0.2222222
2	{Vpeak25=[94,100)}	0.2222222
3	{SaO2=[95,96)}	0.2222222
4	{Age=[30,54)}	0.2777778
5	{Age=[54,61)}	0.2777778
6	{Height=[161,168)}	0.2777778
7	{CD4=[26,28), CD4/CD8=[0.81,1.06)}	0.1111111
8	{6min_test_walk=[365,420), CD4=[26,28)}	0.1666667
9	{CD4=[26,28), CD8=[21,25)}	0.1111111
10	{Force_exhalation_volume=[100,105),CD4=[26,28)}	0.1111111
11	{CD4=[26,28), TNF-α=[11.7,27.3]}	0.1111111
12	{CD4=[26,28), IL-10=[3.7,7.83)}	0.1111111
13	{CD4=[26,28), IL-8=[43.8,98.1]}	0.1111111
14	{Weight=[59,75.7), CD4=[26,28)}	0.1111111

The summary of feature selection by different methods

Feature selector	Features list	Weighted list
Features without high correlation	Age BMI CAT Pulse 6 min test walk Sa02% Borg scale Force lung capacity Force exhalation volume Volume of peak flow at 25% (Vpeak25) Volume of peak flow at 50% (Vpeak50) Volume of peak flow at 75% (Vpeak75) CD16 IL-8 IL-10 CD4/CD8	no
Decision tree (CART)	Force lung capacity Force exhalation volume Vpeak25 Vpeak50 Vpeak75 CD16 IL-8 CD4/CD8	yes
Random forest	Force lung capacity Force exhalation volume Vpeak25 Vpeak50 CD16 CD4/CD8 Vpeak75	yes
Boruta	Force lung capacity Force exhalation volume Vpeak25 Vpeak50 Vpeak75 0-lymphocytes IL-8	yes

The post-COVID rehabilitation duration prediction using different ML models

Whole dataset

Model	AUC	CA	F1	Precision	Recall
Tree	0.854	0.760	0.762	0.766	0.760
SVM	0.988	0.910	0.921	0.924	0.920
Naive Bayes	0.957	0.860	0.861	0.869	0.860
Calibrated Learner	0.917	0.920	0.921	0.933	0.920
Logistic Regression	0.898	0.800	0.800	0.867	0.800
Three-layer stacking ensemble classification model with Random forest aggregate	0.992	0.930	0.960	0.964	0.960

Selected features

Model	AUC	CA	F1	Precision	Recall
Tree	0.781	0.720	0.723	0.735	0.720
SVM	0.908	0.840	0.842	0.847	0.840
Naive Bayes	0.883	0.860	0.861	0.869	0.860
Calibrated Learner	0.888	0.860	0.861	0.896	0.860
Logistic Regression	0.880	0.840	0.841	0.886	0.840
Three-layer stacking ensemble classification model with Random forest aggregate	0.978	0.920	0.921	0.924	0.920

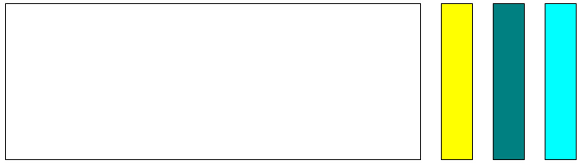
Improved stacking

1) Weak predictors training

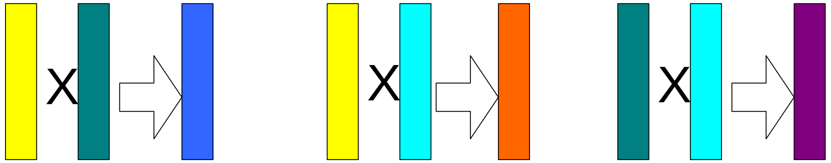
$$\{z_1^1, \ldots z_B^1\}, \{z_1^2, \ldots z_B^2\} \ldots, \{z_1^K, \ldots z_B^K\}$$

Training dataset (with cross-validation)

Results on training dataset

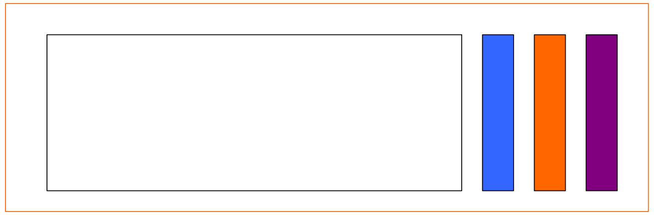


2) Metafeatutres deformation using pairwise multiplication

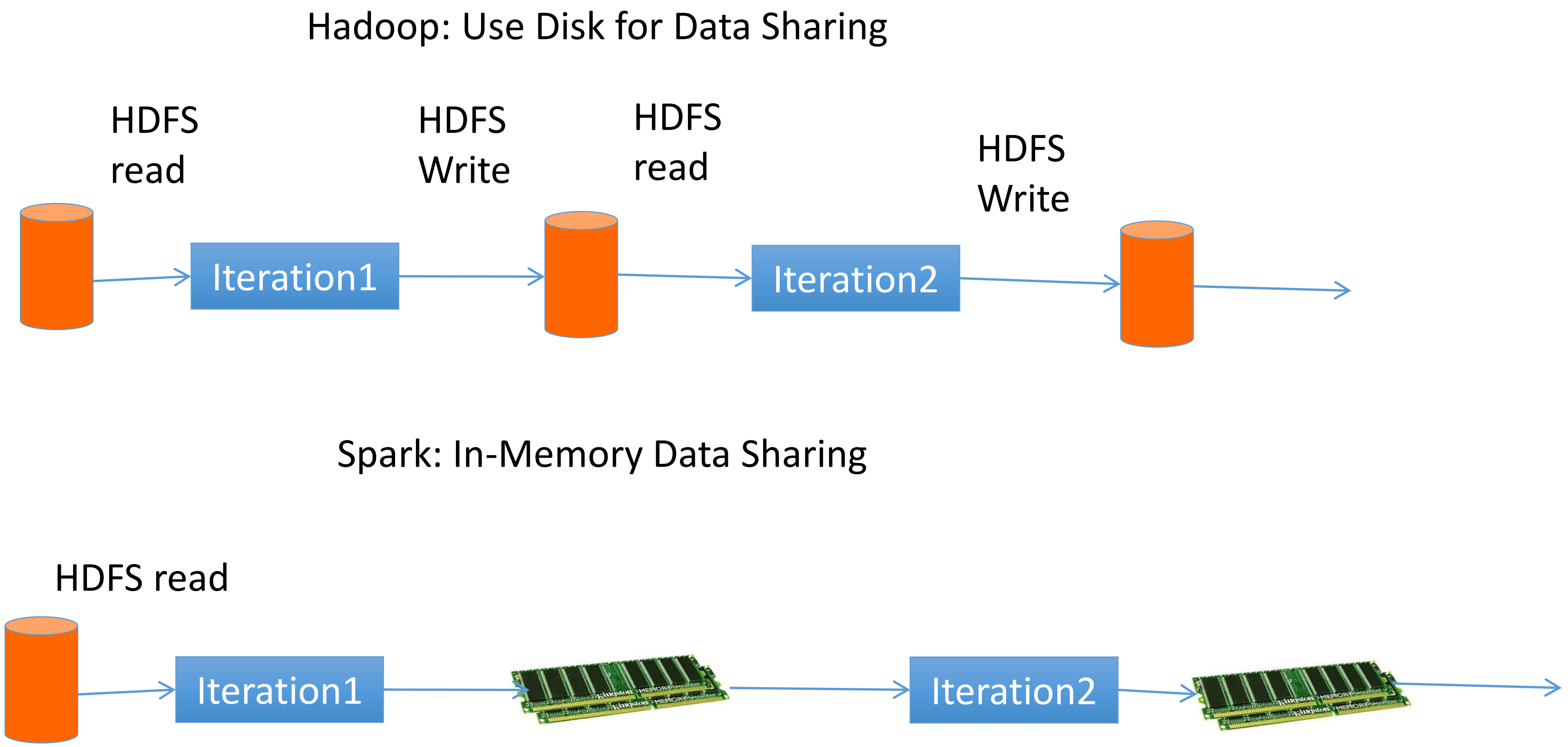


$$s_K(.) = mw\big(w_1(.) \times w_2(.), w_1(.) \times w_3(.), \ldots, w_{K-1}(.) \times w_K(.)\big)$$

3) Meta-algorithm training: initial dataset with deformated metafeatures



Spark Uses Memory instead of Disk



Problem formulation

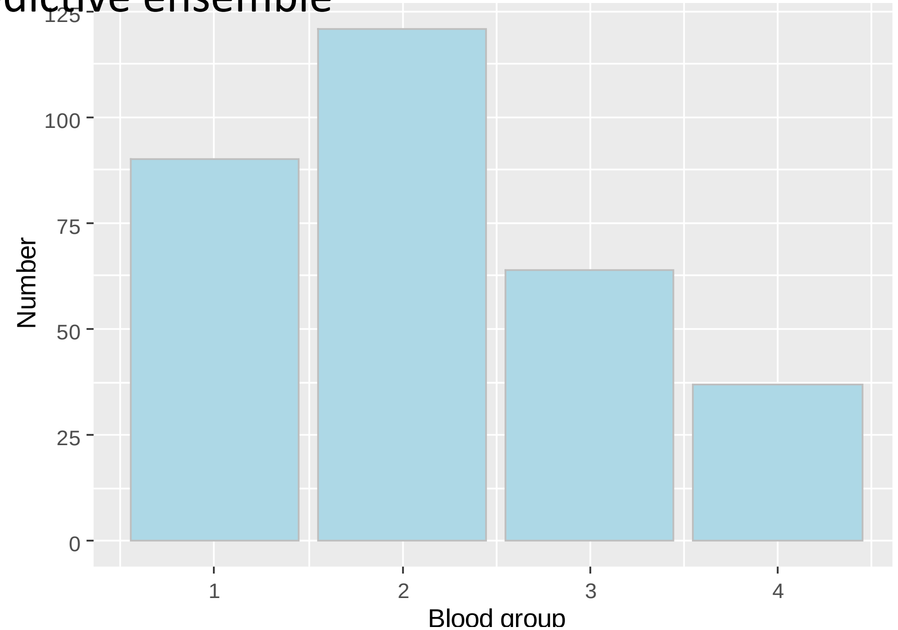
- To find frequent patterns
- To find parameters affected by COVID-19

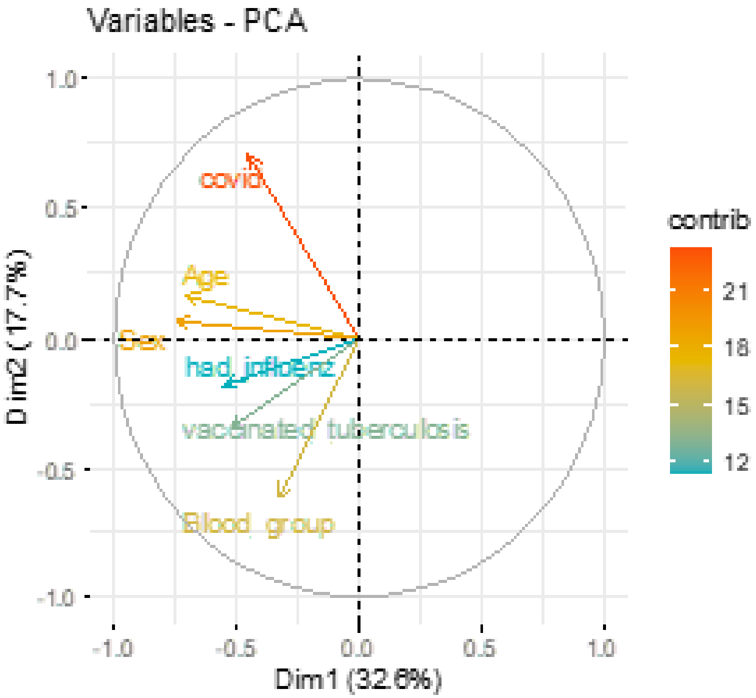
Dataset description

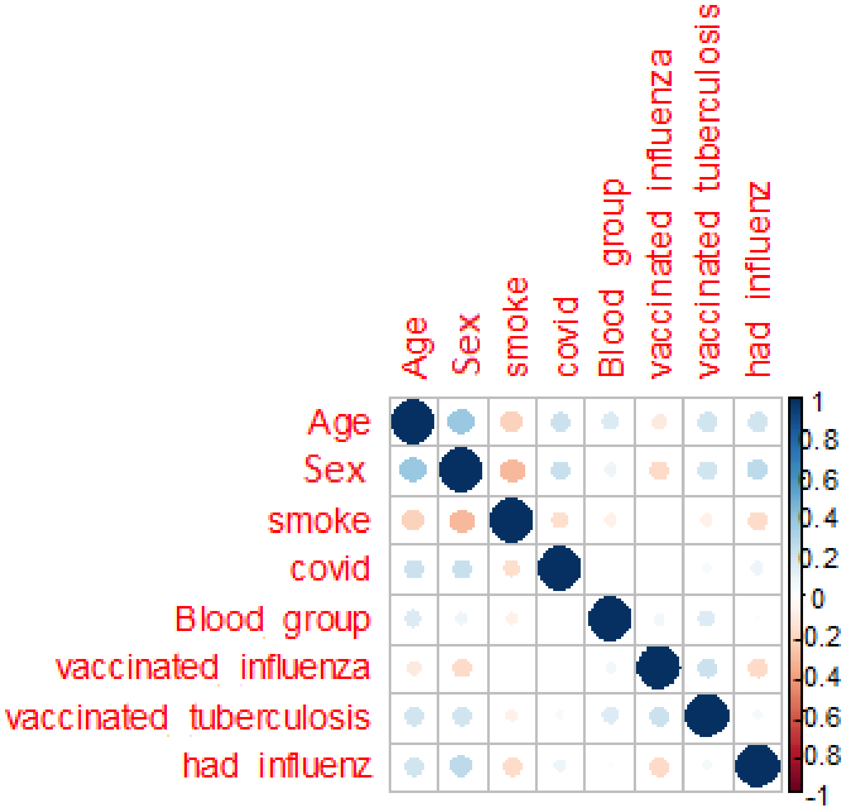
- Dataset is collected under supporting of Central European Initiative and verified by Lviv regional center COVID-19 resistance
- The project Stop COVID-19 has use case, implemented in Ukraine and Belarus. Partners from Germany shared google form too. Dataset is collected data over the period from September 01, to October 29.
- The dataset provides data of COVID-19 unconfirmed and confirmed cases
  - Age (categorical): 1:<15, 2: 16-22, 3: 23-40, 4: 41-65,5: >66,
  - Sex (categorical): male, female,
  - Region (string): Lviv (Ukraine), Chernivtsi (Ukraine), Belarus, Germany, Other,
  - Do you smoke (Boolean): 2:yes, 0: no,
  - Have you had COVID (categorical): 2: yes, 0: no, 1: maybe,
  - IgM level (numerical): [0..0.9) (negative), [0.9..1.1) (indefinite), >=1.1 (positive),
  - IgG level (numerical): [0..0.9) (negative), [0.9..1.1) (indefinite), >=1.1 (positive),
  - Blood group (categorical): 1, 2, 3, 4,
  - Do you vaccinated influenza? (categorical): 2:yes, 0:no, 1:maybe,
  - Do you vaccinated tuberculosis? (categorical): 2:yes, 0:no, 1:maybe,
  - Have you had influenza this year? (categorical): 2:yes, 0:no, 1:maybe,
  - Have you had tuberculosis this year? (categorical): 2:yes, 0:no, 1:maybe.

Predictive ensemble

Preprocessing stage

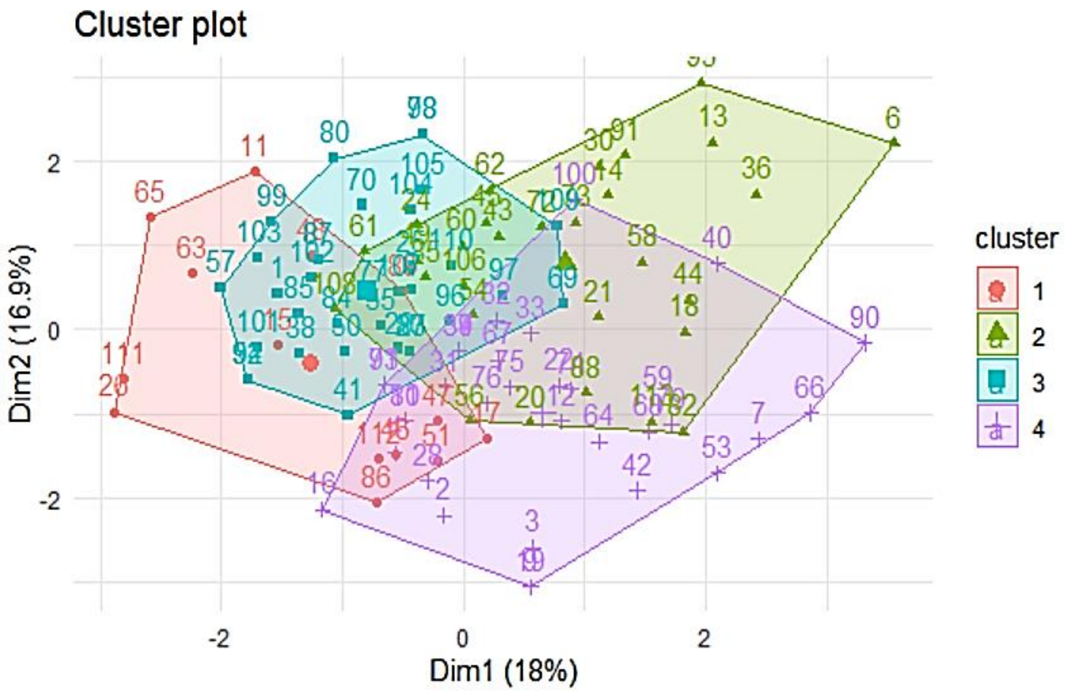


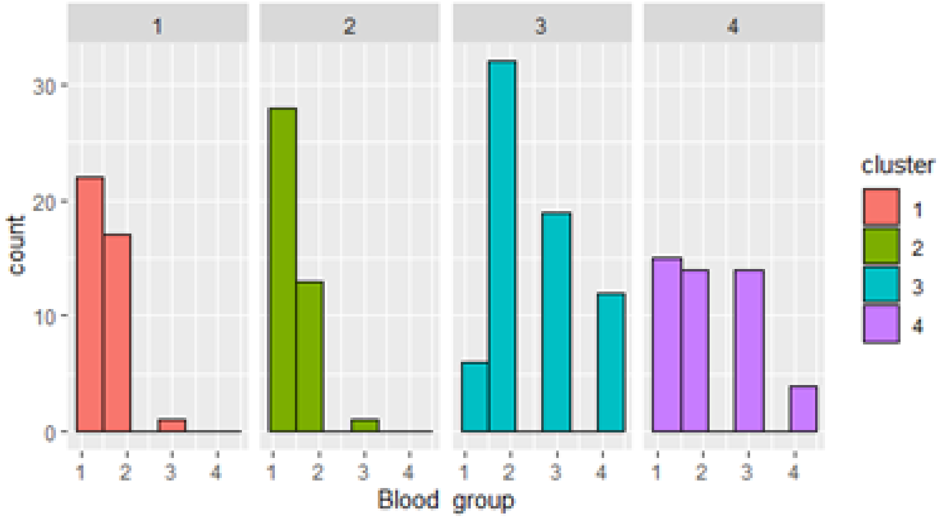


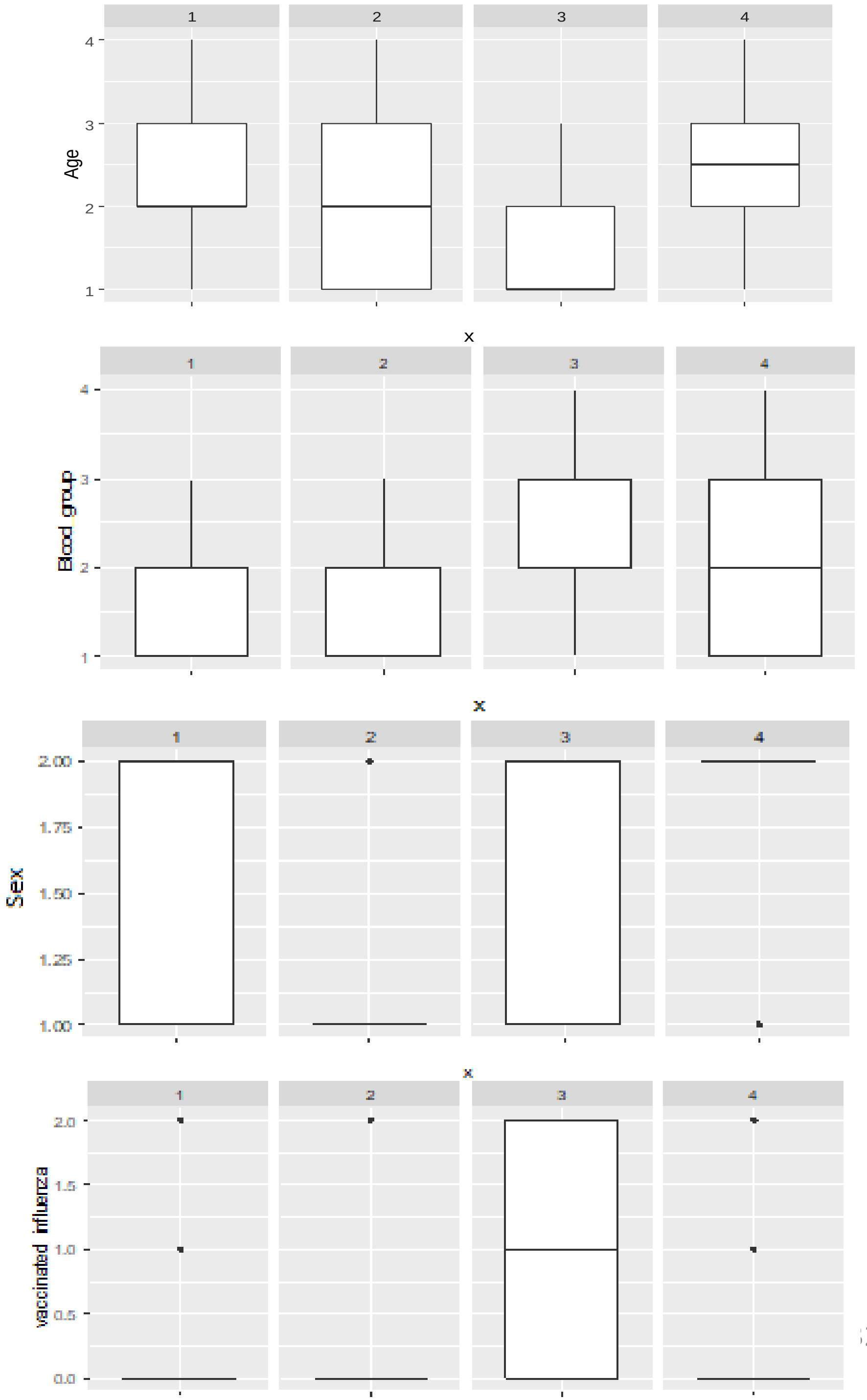


Predictive ensemble

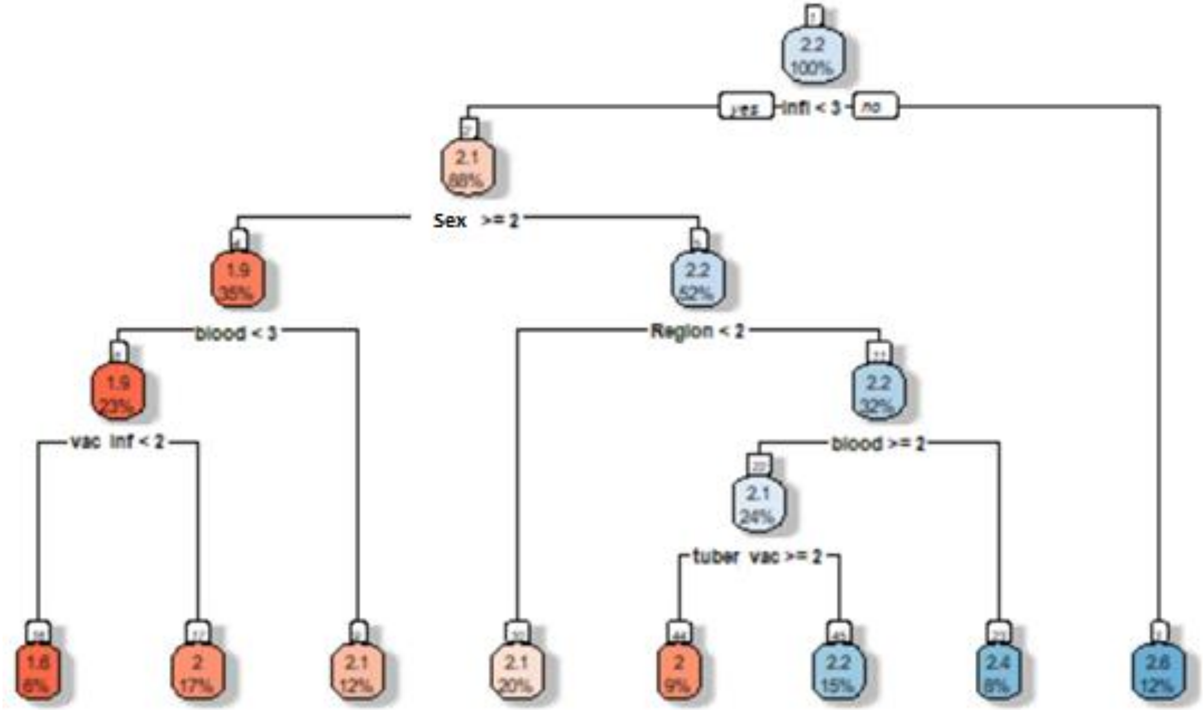
Clustering







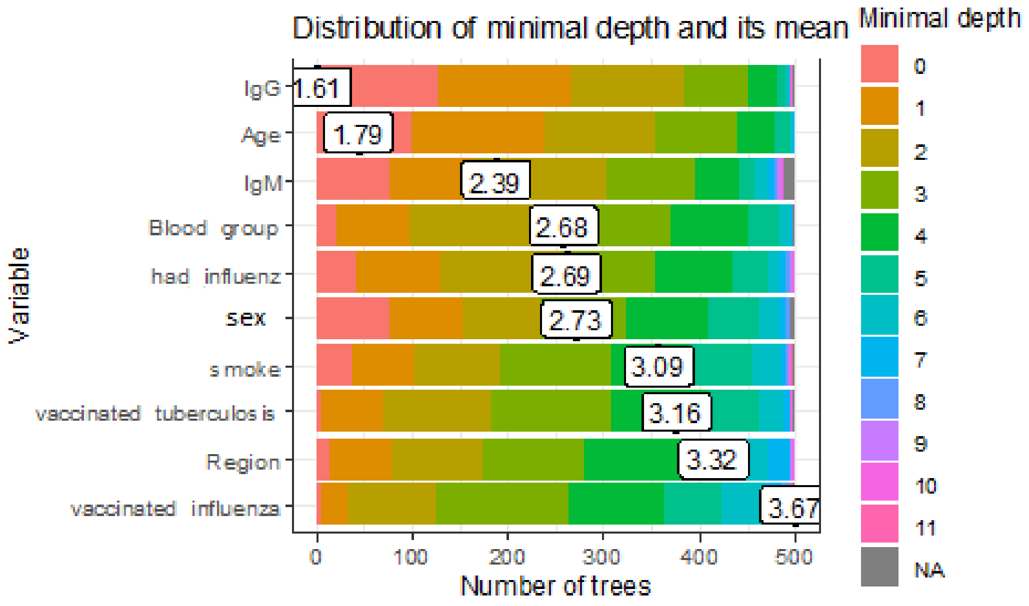
Classification



The accuracy is equal to 0.5135. But, this model allows choosing the main features as following "Have you had influenza this year", Sex, blood group, region.

RF - Confusion matrix

	0	1	2	Class error
0	153	9	17	0.14525139
1	8	88	12	0.18518518
2	1	3	20	0.16666666



Models' accuracy for whole features

Model	Full dataset	Filtered by Ukraine	Filtered by Belarus	Filtered by Germany	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Logistic regression	0.553	0.572	0.534	0.544	0.601	0.592	0.610	0.589
Support vector machine	0.605	0.6327	0.570	0.584	0.621	0.694	0.635	0.637
Naive Bayes	0.670	0.693	0.655	0.655	0.674	0.693	0.672	0.692
XGBoost	0.898	0.932	0.860	0.942	0.941	0.945	0.899	0.957
Random Forest	0.897	0.924	0.859	0.940	0.932	0.944	0.961	0.925
Neural network	0.820	0.849	0.828	0.79	0.830	0.849	0.8204	0.849
Decision tree	0.513	0.542	0.517	0.492	0.553	0.631	0.612	0.642

Models' accuracy for selected features

Model*	Age, IgG, Blood_group, had_influenz, IgM	Age, Sex, Blood_group, had_influenz
Logistic regression	0.633	0.671
Support vector machine	0.671	0.722
Naive Bayes	0.674	0.732
XGBoost	0.935	0.945
RandomForest	0.945	0.934
Neural network	0.832	0.845
Decision tree	0.553	0.631

Hierarchical classifier is built as following

1. Using gaps-statistics the appropriative number of clusters is found. This number is equal to four;
2. k-means divides objects by 4 groups; density of distribution is calculated;
3. Weak predictors are used for each cluster separately. The best predictors choosing;
4. Improved stacking is used.

Results

The accuracy of hierarchical classifier

Model	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Hierarchical classifier	0.941	0.945	0.961	0.957

Department of artificial intelligence. Lviv Polytechnic National University

- The youngest department (5 years)
- Appr. 500 students (bachelor, master, PhD)
- Specialty “Computer sciences”, specialization “Artificial intelligence”

Mission of the department: to grow highly motivated professionals whose ideas would form vectors for the development and use of artificial intelligence to solve problems with social and economic impact

Statistics:

- 32 scientists and teacher staff (7 professors, 18 PhD, 32 PhD students)
- More than 50 publications in international journals scientometric databases (Scopus, Web of Science) with impact-factor and Q>=2,
- 11 patents,
- 20 projects (6 international projects, 4 DAAD projects, 5 projects for private organizations and public authorities, 2 project funded by Ukrainian National Research Foundation, 3 Ukrainian projects)
- organizer of International conference “Informatics and data-driven medicine”, ranked in Core (rank C, http://portal.core.edu.au/conf-ranks/2276/ – this conference is only one Ukrainian conference ranked in this category).

Education process

- Join programs with Lviv IT-cluster for bachelor and master students
- Join degree Master program with Lviv IT-cluster, Slovak IT association and University of Bratislava
- Join degree Master program with Wurzburg university, Germany
- Dual education with SoftServe IT company
- Success PoC with GlobalLogic IT company

Scientific research and international cooperation – projects

Compiled International projects

- Horizon2020 project for cascade funding: “Hub laboratory Internet of things” (https://s3platform.jrc.ec.europa.eu/digital-innovation-hubs-tool/-/dih/1472/view)
- Central European Initiatives: Stop Covid-19 (https://www.cei.int/news/8992/powering-data-driven-actions-in-fight-against-covid-19-in-Ukraine)
- Wurzburg university: The development of neural controller for small satellite rotation
- Lectura (Germany): Gap filling and semi structural data analysis
- USA company: Behavior analysis

In progress

- Horizon2020 project: AURA - aurization of opera houses and concert halls
- EUREKA: Integrated Care for Next Generation

Scientific research and international cooperation – projects

Compiled national projects

- Lviv regional administration: An electronic queue at kindergartens in Lviv region
- Biofarma: National registry of patients with imunodeficits
- Ukrainian-German enterprise Spheros-Electron: Industrial IoT solution (hardware-software system)
- Ukrainian company: Chat-bots army
- Social media group: Propaganda recognition

In progress

- Ukrainian company: automatically documents recognition for insurance company

Scientific research and international cooperation – research

- Data augmentation and Missing data imputation
- Cellular automata and genetic algorithms
- Big data analysis
- Small data analysis
- Text mining (emotion recognition)
- Pattern recognition (probabilistic dependencies, neuro-fuzzy approach)
- Software quality analysis

Students project:

- AR & VR
- robotics