Distribution and Replication for Feature Selection

(ESR 2.2)

Uchechukwu Njoku

eBISS 2022
Cesena, Italy
8th July, 2022

Supervisors: Alberto Abelló (UPC), Besim Bilalli (UPC), Gianluca Bontempi (ULB)

Feature selection (FS)

Feature selection is the process of **detecting the relevant features and discarding the irrelevant and redundant ones** with the goal of obtaining a subset of features that accurately describe a given problem with a **minimum degradation of performance**

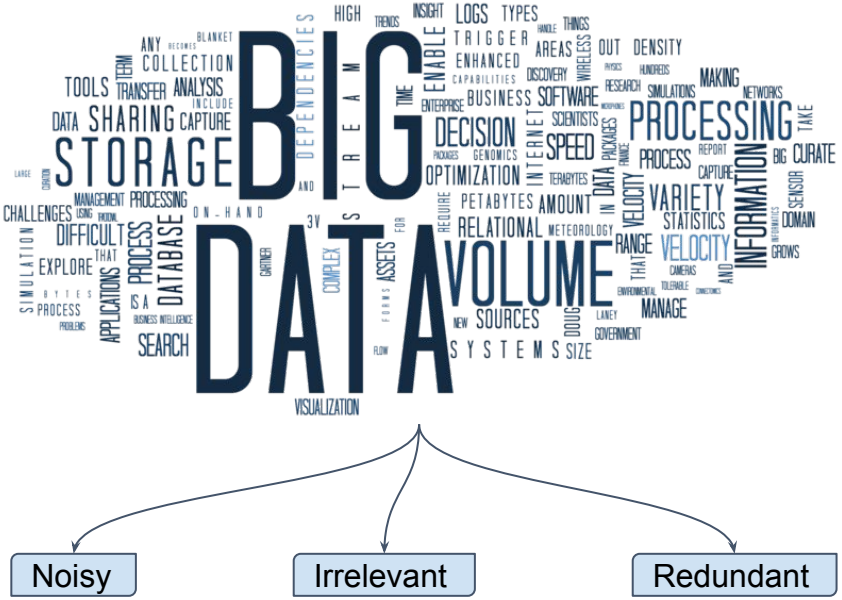
F_1	F_2	F_3	…	F_m	…	F_n
c_{11}	c_{12}	c_{13}	…	c_{1m}	…	c_{1n}
c_{21}	c_{22}	c_{23}	…	c_{2m}	…	c_{2n}
…	…	…	…	…	…	…
c_{j1}	c_{j2}	c_{j3}	…	c_{jm}	…	c_{jn}

Fig. 1: Full dataset

F_1	F_2	F_3	…	F_m
c_{11}	c_{12}	c_{13}	…	c_{1m}
c_{21}	c_{22}	c_{23}	…	c_{2m}
…	…	…	…	…
c_{j1}	c_{j2}	c_{j3}	…	c_{jm}

Fig. 2: Filtered dataset

Why feature selection?



- Storage
- Training time
- Data visualization
- Data understanding
- Curse of dimensionality

FS classification

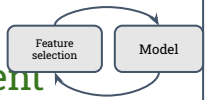
Feature selection methods are popularly classified based on their relationship with the learning algorithm

Filter methods

- Fast execution
- Good generalization
- Robust to overfitting
- Possible redundancy
- Model independent
- Non 'optimal' selection

E.g: Gini, ReliefF, MRMR, CFS

Wrapper methods



- Model dependent
- High accuracy
- Captures dependencies
- Poor generalization
- Risk of overfitting
- Computationally intense

E.g: SFS, SBS

Embedded methods

- Model dependent
- Moderate execution
- Captures dependencies
- Poor generalization

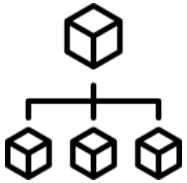
E.g: Tree based algorithms

Challenges of FS [1, 2]

- Scalability
- Feature cost
- Interpretability
- Distributed data
- Reduced-Precision
- Need for ensembles
- Millions of dimensions
- Instance-Based (use-case based)
- Incremental data (real-time, online)

Focus

- Scalability
- Feature cost
- Interpretability
- Distributed data
- Reduced-Precision
- Need for ensembles
- Millions of dimensions
- Instance-Based (use-case based)
- Incremental data (real-time, online)



Related Works

Challenge	Methodology	Reference
Scalability and Distributed data	MapReduce • Filter • Wrapper	 [5] [3]
	Spark • Filter • Wrapper	 [8, 11, 12, 15, 16] [6, 17]
	GPU • Filter	 [11]
	MPI • Filter	 [10]
	Vertical partitioning • Filter • Wrapper	 [7, 8, 9, 13, 14, 18] [4]
	Horizontal partitioning • Filter • Wrapper	 [8, 9, 13, 18] [6]
	Early removal/dropping • Filter	 [18, 19]

Gaps

- Focus on filter methods
- Single criteria selection methods
- Proposed methods not accompanied with tools
- Insufficient post selection insight for explainability

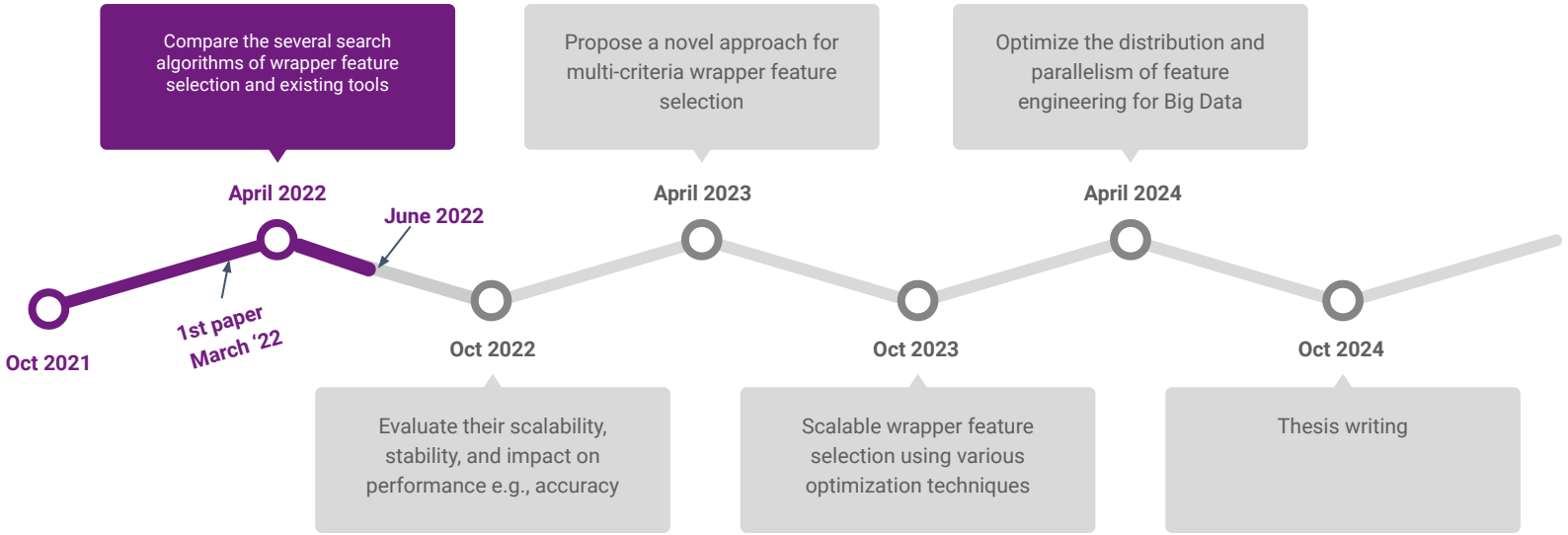
Goal

This project focuses on making **wrapper feature selection** methods **scalable** by optimizing their search strategies through distribution and parallelism to enhance data preprocessing and, consequently, **improve the performance** of learning algorithms

Objectives

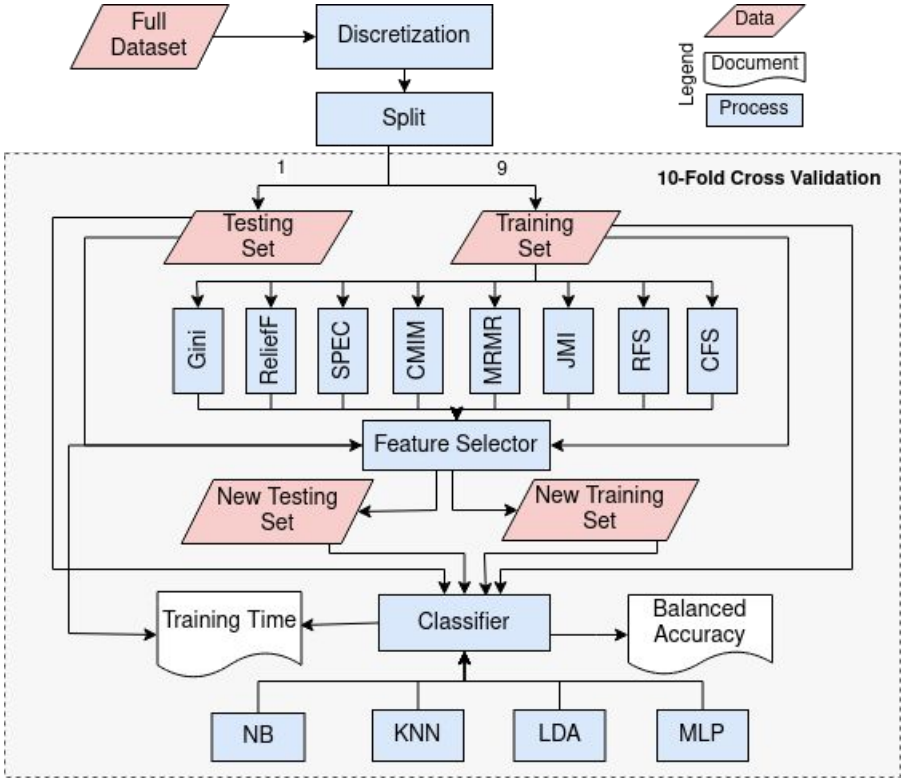
1. Study existing feature selection methods to:
   a. compare the several search algorithms of wrapper feature selection and existing tools
   b. evaluate their scalability, stability, and impact on performance e.g., accuracy
2. Propose a novel approach for **multi-criteria** wrapper feature selection
3. Optimize wrapper feature selection methods by adopting frameworks for distribution, parallel computing, load partitioning, and communication methods for **scalable** feature selection
4. **Optimize** the distribution and parallelism of feature engineering for Big Data
   a. Analyze the scalability of feature engineering methods

Timeline


Compare the several search algorithms of wrapper feature selection and existing tools
Propose a novel approach for multi-criteria wrapper feature selection
Optimize the distribution and parallelism of feature engineering for Big Data
April 2022
June 2022
April 2023
April 2024
Oct 2021
1st paper March '22
Oct 2022
Oct 2023
Oct 2024
Evaluate their scalability, stability, and impact on performance e.g., accuracy
Scalable wrapper feature selection using various optimization techniques
Thesis writing

Impact of filter feature selection on classification: an empirical study [19]

Methodology



Work Load

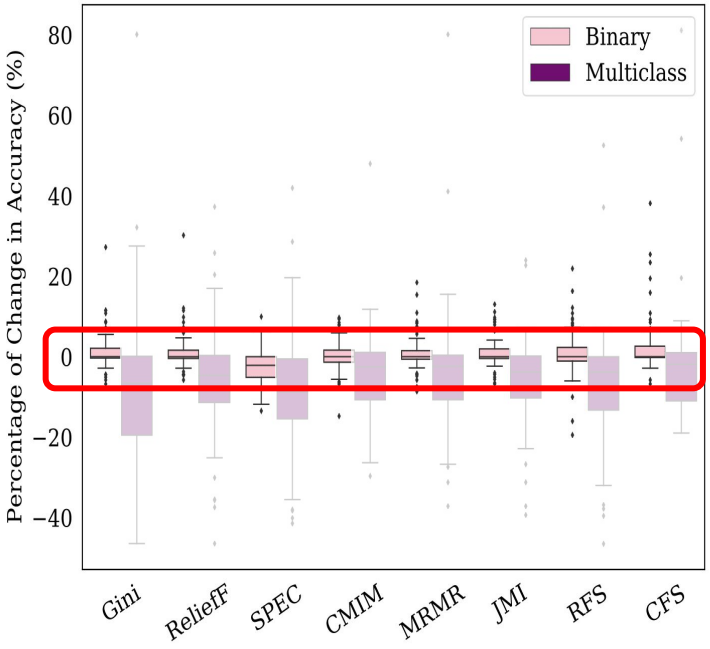
- 32 datasets
- 8 feature selection methods
- 4 classification algorithms
- 5 feature subset sizes [#features^(0.5,0.6,0.7,0.8,0.9)]

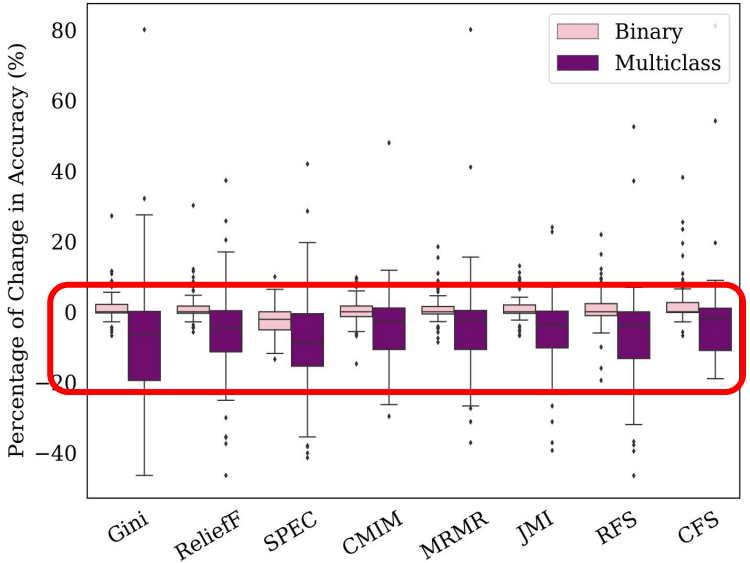
Metrics

- Feature selection runtime
- Classifier accuracy change
- Classifier runtime change

Classifier accuracy

The improvement in accuracy after feature selection is different for binary and multiclass classifications





Current focus

Objective 1: Study present feature selection methods

Aims:

- Show the limitation of current tools
- Define benchmark for comparison with proposed solution

Tools for Wrapper FS: specifications

Tool	Lang.	Exponential	Population-based	Sequential	Parallelism
Weka	Java	X	X	✓	Partial
Sklearn	Python	X	X	✓	Partial
Rapidminer	Java	✓	✓	✓	Partial
Mlxtend	Python	✓	X	✓	Yes
scikit-feature	Python	X	X	✓	No
FeatureSelect	Matlab	X	✓	X	No

Partial means it is not available for all algorithms

Limitations

Tool	Limitations
Weka	Poor implementation documentation
Sklearn	Naive implementation of sequential selection (single stop criterion)
Rapidminer	RapidMiner Studio Free upto 10,000 data rows and 1 Logical Processor
Mlxtend	Naive implementation of sequential selection (single stop criterion)
scikit-feature	Naive implementation of sequential sequential feature selection with evaluation criteria (Support Vector Machine and Decision Tree)
FeatureSelect	Not maintained

Setup

Dataset
- Binary
- 16 real
- One synthetic

Classification algorithms
- Naive Bayes
- K-Nearest Neighbour

FS methods
- Wrapper-(Sequential Forward Selection)
- Filters-(Five methods)

Evaluation

Internal
- Accuracy change
- AUC change

External
- Stability
- Execution time
- Peak memory usage

Accuracy/AUC change

- Base classifier
- New Classifier
- New > Base

$$\frac{\text{New - Base}}{\text{Base}}$$

NB - Accuracy change (%)

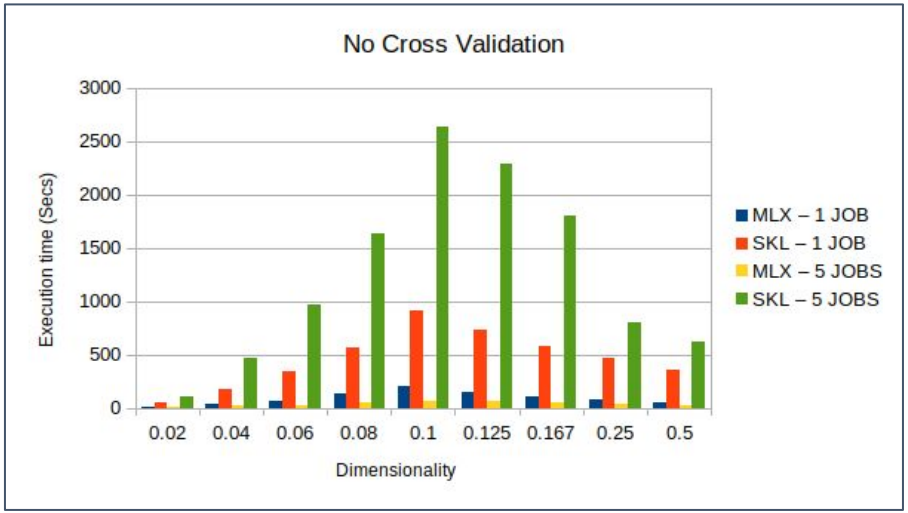
Dataset	SFS	Gini	ReliefF	JMI	CMIM	mRMR	SPEC
confidence	1.231	-0.577	-0.577	**1.354**	**1.354**	-0.577	-0.577
fri_c3_250_10	10.659	**31.707**	**31.707**	**31.707**	13.916	13.916	-34.811
page-blocks	**4.550**	-0.455	-2.097	-1.579	-1.579	-0.455	-0.015
delta_elevators	-0.408	**0.632**	**0.632**	**0.632**	**0.632**	**0.632**	-32.247
synthetic_control	**0**	**0**	-1	**0**	**0**	**0**	-0.465
isolet	**4.339**	-0.202	0.067	0.051	-1.813	-0.119	-15.774
mfeat-zernike	**7.123**	-0.495	-0.295	-0.355	-0.330	-0.295	-10.634
gina	**7.756**	2.140	4.745	2.476	4.647	5.864	-20.622
blood-transfusion	**1.351**	-1.114	-0.658	-1.114	-1.114	-1.114	-1.773
disclosure_z	0.892	5.188	0.495	5.188	5.188	**6.865**	5.188
wilt	**6.020**	-3.968	0.132	-3.968	-3.968	-3.863	-3.772
stock	**10.400**	-1.156	-1.842	1.523	-1.842	-2.449	-6.553
ar4	**2.080**	1.053	0	-0.721	0.040	-0.112	1.164
fri_c4_250_100	14.621	**36.266**	23.366	24.938	16.962	28.876	-2.957
clean2	**12.500**	0.885	0.426	0.803	0.372	0.478	-3.789
philippine	5.655	**17.568**	0.021	4.056	4.916	1.667	-1.744
Count	10	5	2	4	3	3	0

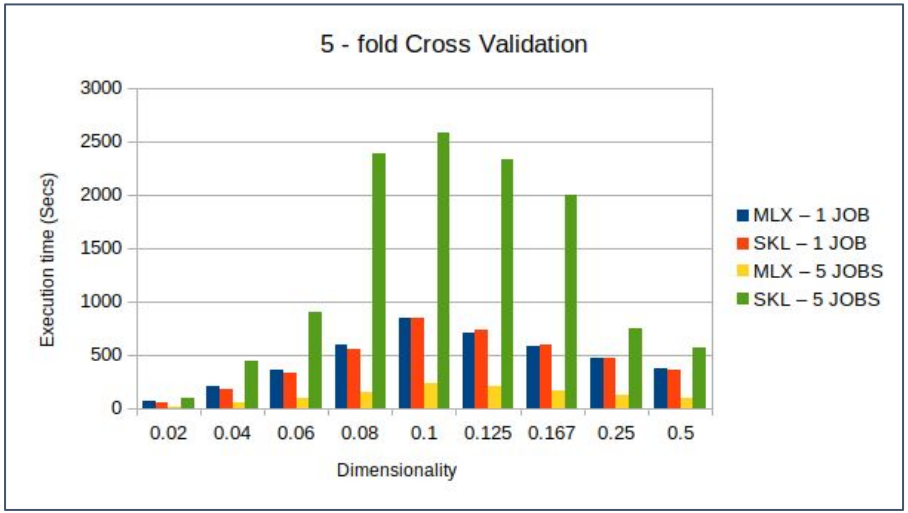
Accuracy change - Ranking

Method	NB_ACC	NB_AUC	KNN_ACC	KNN_AUC	Avg. rank
SFS	10	5	9	13	1
Gini	5	7	2	3	2
ReliefF	2	2	1	4	6
JMI	4	3	4	2	4
CMIM	3	3	2	1	5
mRMR	3	5	3	3	3
SPEC	0	2	2	2	7

Re-establishing the need for wrapper FS

Scalability





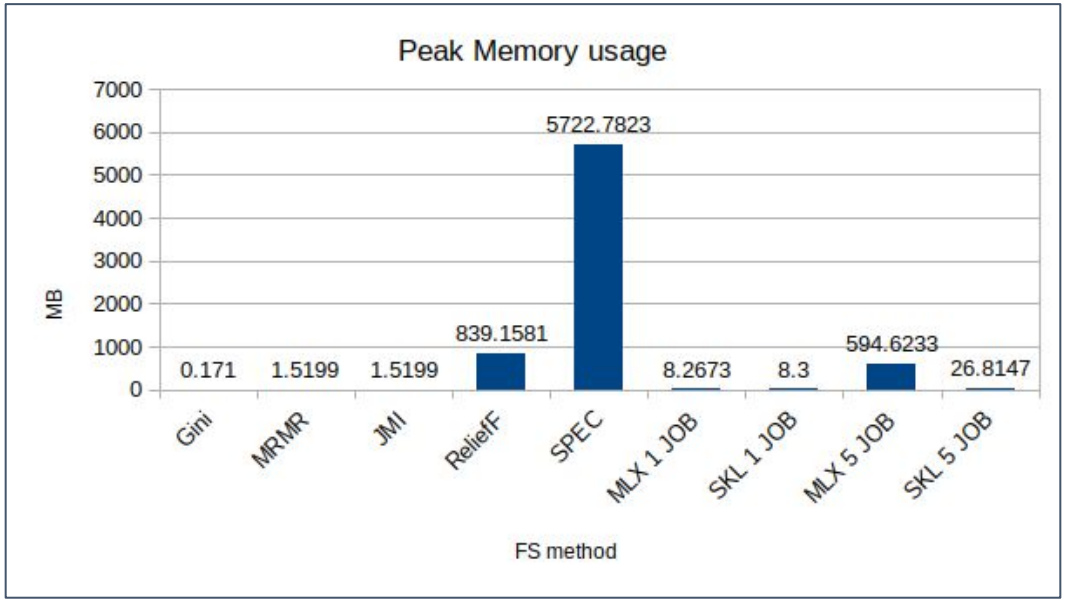
Stability

Dataset	KNN-AC	NB-ACC	NB-AUC	Gini	ReliefF	JMI	CMIM	mRMR	SPEC
confidence	1	0.4667	0.6614	1	1	1	1	1	1
fri_c3_250_10	1	0.746	0.7778	1	1	1	1	1	1
page-blocks	1	1.00E+00	1	1	1	1	1	1	1
delta_elevators	1	1	0.1635	1	1	1	1	1	1
synthetic_control	0.2276	2.95E-01	0.0347	1	1	1	1	1	1
isolet	0.1162	0.055	0.5897	1	1	1	1	1	1
mfeat-zernike	0.2913	0.5934	0.5327	1	1	1	1	1	1
gina	0.1625	0.2528	0.4667	1	1	1	1	1	0.004
blood-transfusion	-0.0444	1	1	0.4667	1	0.4667	0.4667	0.4667	1
disclosure_z	0.3	1	0.6111	1	1	1	1	1	1
wilt	0.4259	0.0185	1	1	1	1	1	1	1
stock	1	1	0.2965	1	1	1	1	1	1
ar4	0.1569	0.0763	0.4198	1	1	1	1	0.8711	1
fri_c4_250_100	0.3753	0.2914	0.2719	1	1	1	1	1	1
clean2	0.5405	0.3182	0.587	1	1	0.9018	0.7888	1	1
philippine	0.1661	0.5765	0	0.941	1	0.8282	0.7312	0.8767	0.0114

Wrappers ⟷ Filters

Peak memory usage

At 1,000 features and 10,000 instances

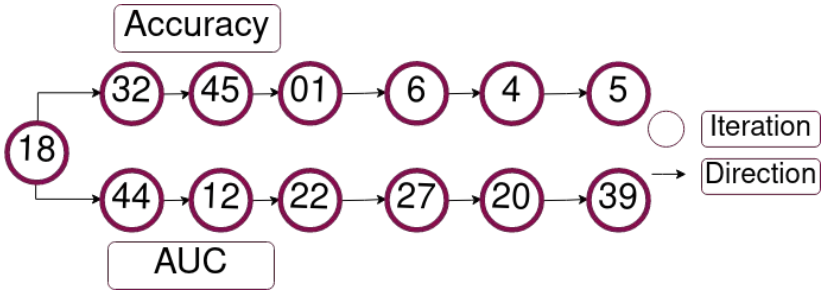


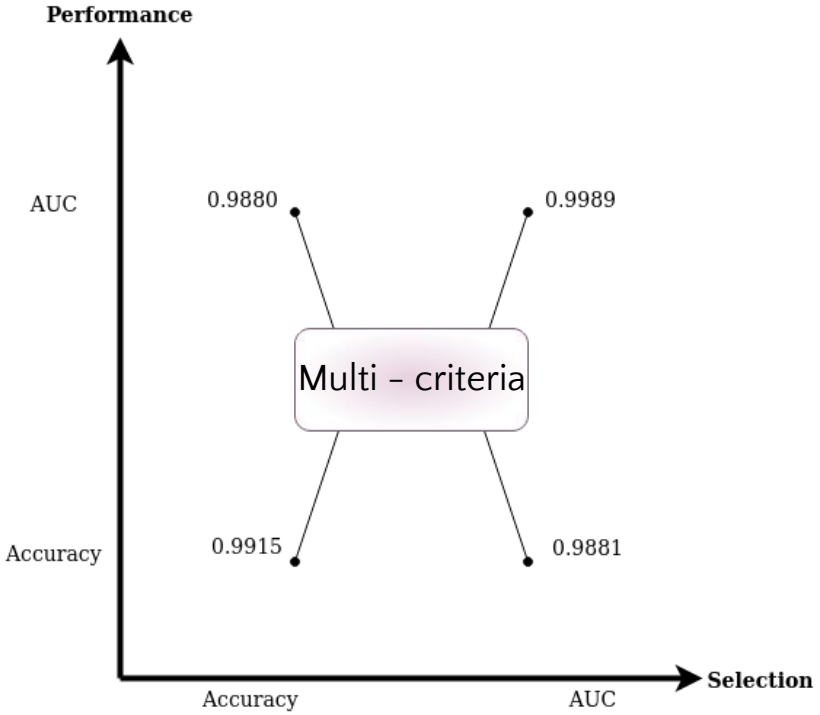
Selection criteria

Given a binary dataset:

- 48 features
- 2,000 instances

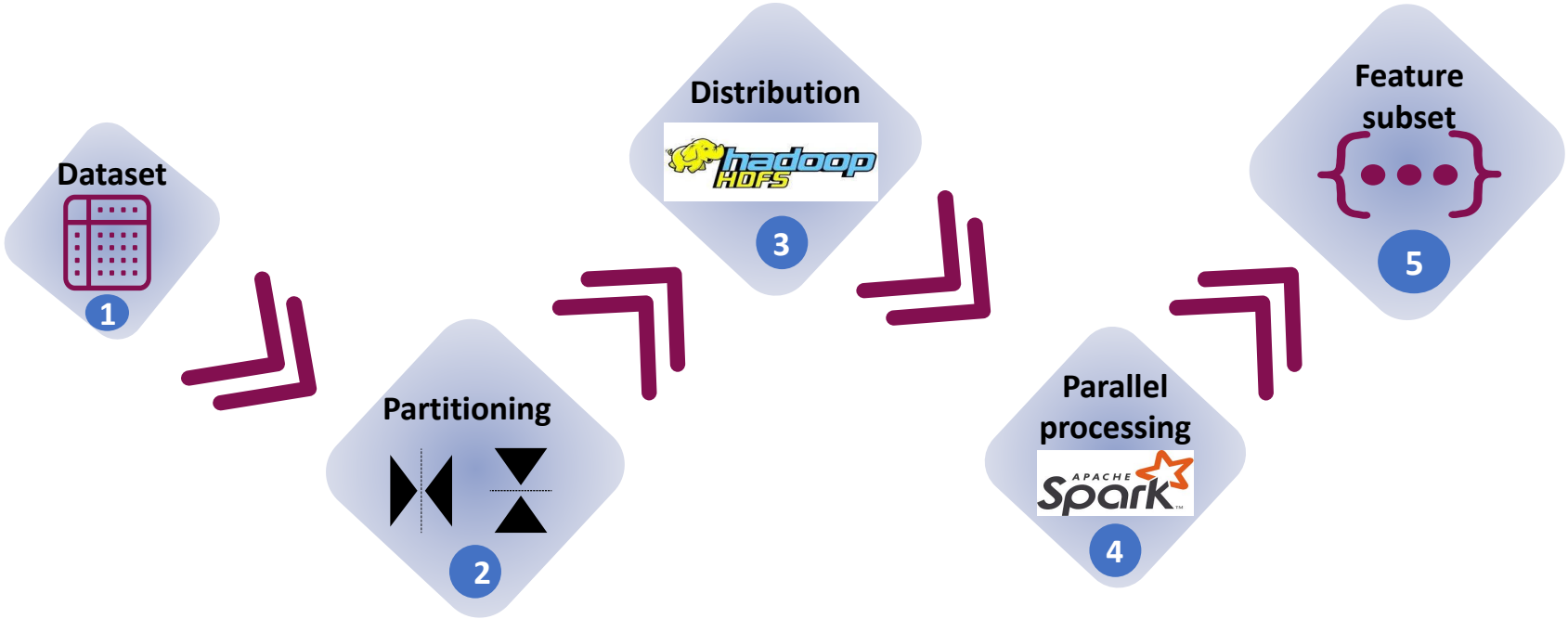
We select the **seven** most informative features.





Next Steps

Propose a novel approach for **multi-criteria** wrapper feature selection



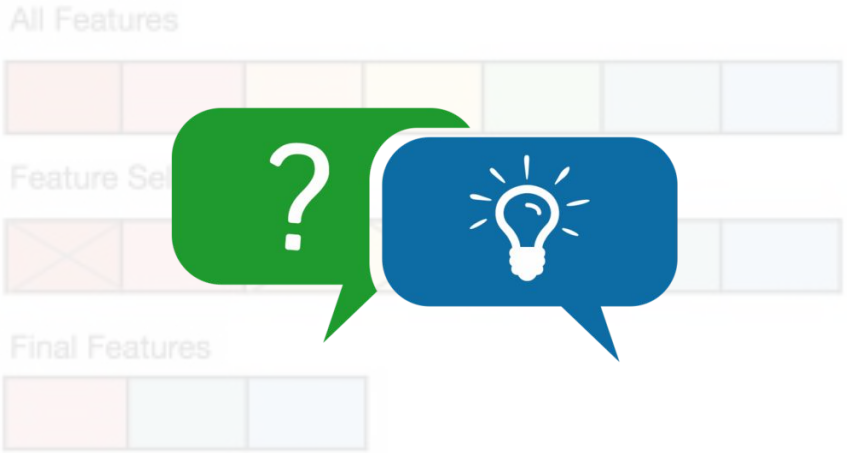
Project outcome

A feature selection tool that provides a suite of methods with various metrics for selection and post-selection evaluation to enable faster feature selection, interpretability, and explainability.

References

1. Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data
2. Bolón-Canedo, V., Alonso-Betanzos, A., Morán-Fernández, L., & Cancela, B. (2022). Feature Selection: From the Past to the Future.
3. Keco, D., & Subasi, A. (2012). Parallelization of genetic algorithms using Hadoop Map/Reduce
4. Bolón-Canedo, V., Sánchez-Marono, N., & Alonso-Betanzos, A. (2013, April). A distributed wrapper approach for feature selection
5. Sun, Z. (2014). Parallel feature selection based on MapReduce
6. Peralta, D., del Río, S., Ramírez-Gallego, S., Triguero, I., Benitez, J. M., & Herrera, F. (2015). Evolutionary feature selection for big data classification: A mapreduce approach
7. Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Distributed feature selection: An application to microarray data classification
8. Palma-Mendoza, R. J., de-Marcos, L., Rodriguez, D., & Alonso-Betanzos, A. (2019). Distributed correlation-based feature selection in spark
9. Bolon-Canedo, V., Sechidis, K., Sanchez-Marono, N., Alonso-Betanzos, A., & Brown, G. (2019). Insights into distributed feature ranking
10. González-Domínguez, J., Bolón-Canedo, V., Freire, B., & Touriño, J. (2019). Parallel feature selection for distributed-memory clusters
11. Ramírez-Gallego, S., Lastra, I., Martínez-Rego, D., Bolón-Canedo, V., Benítez, J. M., Herrera, F., & Alonso-Betanzos, A. (2017). Fast-mRMR: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data
12. Ramírez-Gallego, S., Mouriño-Talín, H., Martinez-Rego, D., Bolón-Canedo, V., Benítez, J. M., Alonso-Betanzos, A., & Herrera, F. (2017). An information theory-based feature selection framework for big data under apache spark
13. Morán-Fernández, L., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Centralized vs. distributed feature selection methods based on data complexity measures
14. Zadeh, S. A., Ghadiri, M., Mirrokni, V., & Zadimoghaddam, M. (2017, February). Scalable feature selection via distributed diversity maximization
15. Reggiani, C., Borgne, Y. A. L., & Bontempi, G. (2017, November). Feature selection in high-dimensional dataset using MapReduce
16. Palma-Mendoza, R. J., Rodriguez, D., & De-Marcos, L. (2018). Distributed ReliefF-based feature selection in Spark
17. Galar, M., Triguero, I., Bustince, H., & Herrera, F. (2018, July). A preliminary study of the feasibility of global evolutionary feature selection for big datasets under Apache Spark
18. Tsamardinos, I., Borboudakis, G., Katsogridakis, P., Pratikakis, P., & Christophides, V. (2019). A greedy feature selection algorithm for Big Data of high dimensionality
19. Njoku, U., Abelló, A., Bilalli, B., & Bontempi, G. (2022). Impact of Filter Feature Selection on Classification: An Empirical Study. In DOLAP

Questions & Discussion



Datasets: OpenML Use Case

Number of features:

0: [2, 16] 205 datasets

1: [19, 971] 175 datasets

Number of Instances

0: [27, 846] 246 datasets

1: [937, 9.989] 134 datasets

Classes

0 – Binary 262 datasets

1 – Multiclass 118 datasets

Class balance

0: [0,0385, 0,8083] 84 datasets

1: [0,8232, 1,0] 296 datasets

CFIB

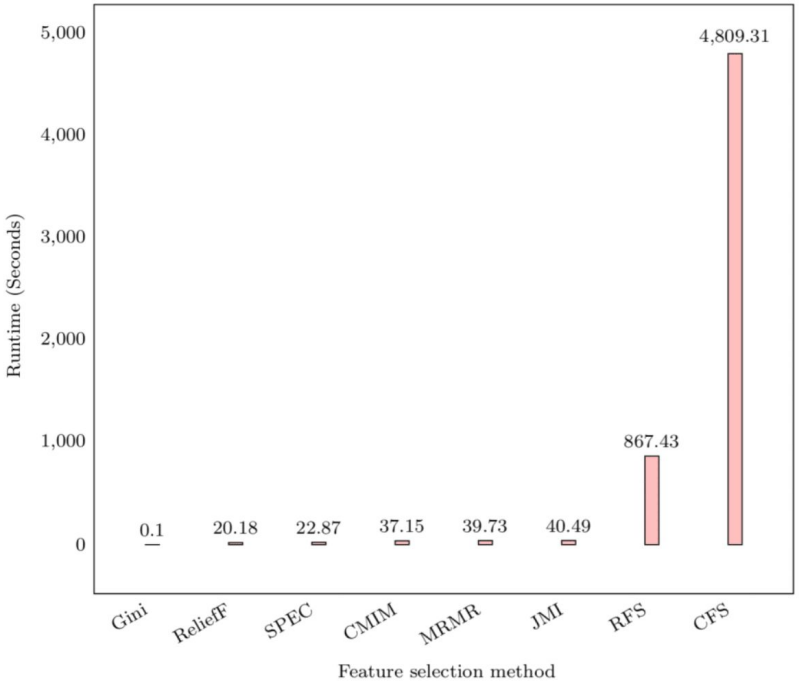
```
0000: 4
0001: 98
0010: 6
0011: 24
0100: 23
0101: 55
0110: 21
0111: 31
```

CFIB

```
1000: 4
1001: 46
1010: 18
1011: 5
1100: 5
1101: 11
1110: 3
1111: 26
```

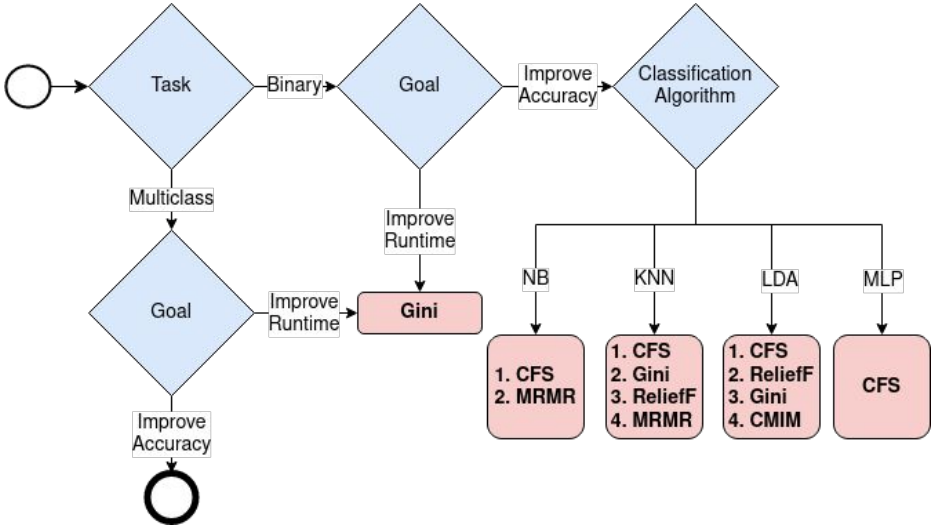
- **All datasets fall under one of 16 (2^4) possible factor combinations**
- **Two representatives chosen randomly**

FS runtime



- Determined by various factors (number of instances, features, class, ...)
- Gini method (less than one second) is most time efficient
- CFS (over 80 minutes) is the least efficient method timewise due to the required computation of pairwise correlations between features
- Selected features can be re-used for model selection

Recommendation



Recommendation based on:

- Goal
- Task
- Classification algorithm

No one size fits all FS method

NB – AUC change (%)

Dataset	SFS	Gini	ReliefF	JMI	CMIM	mRMR	SPEC
confidence	**-0.847**	-6.037	-6.037	-23.932	-23.932	-6.037	-6.037
fri_c3_250_10	**4.624**	-13.317	-13.317	-13.317	-7.931	-7.931	-82.285
page-blocks	3.065	**64.630**	-79.506	36.857	36.857	**64.630**	48.612
delta_elevators	**-0.232**	-0.937	-0.937	-0.937	-0.937	-0.937	-69.911
synthetic_control	-0.001	**0**	-83.491	**0**	**0**	**0**	**0**
isolet	1.096	3.277	3.639	5.684	-11.460	**6.129**	-24.459
mfeat-zernike	1.472	25.933	**33.282**	28.342	25.594	32.006	-37.570
0gina	7.682	2.036	2.192	4.414	6.147	**11.604**	-48.413
blood-transfusion	0.846	**3.611**	-16.356	**3.611**	**3.611**	**3.611**	-100
disclosure_z	0.254	**10.170**	-55.194	**10.170**	**10.170**	-42.793	**10.170**
wilt	**-4.678**	-100	-98.070	-100	-100	-100	-100
stock	4.366	**11.789**	-25.740	4.414	-25.740	-1.911	2.499
ar4	3.341	**9.081**	-3.227	-11.120	-8.461	-7.730	-4.592
fri_c4_250_100	18.815	**84.634**	35.611	33.132	36.314	57.136	-52.705
clean2	8.931	15.034	**47.452**	36.478	39.619	1.626	-34.480
philippine	**5.841**	-1.823	-27.342	-0.832	2.436	5.157	-7.951
Count	**5**	**7**	2	3	3	5	2

KNN – Accuracy change (%)

Dataset	SFS	Gini	ReliefF	JMI	CMIM	mRMR	SPEC
confidence	**1.354**	-0.856	-0.856	-1.563	-1.563	-0.856	-0.856
fri_c3_250_10	**31.707**	2.877	2.877	2.877	3.173	3.173	-17.776
page-blocks	0.606	1.701	-26.390	0.009	0.009	1.701	**3.065**
delta_elevators	**0.632**	-0.232	-0.232	-0.232	-0.232	-0.232	-30.770
synthetic_control	-0.034	**0**	-27.855	**0**	**0**	**0**	**0**
isolet	0.167	0.373	0.714	**0.833**	-1.691	0.602	-6.713
mfeat-zernike	-0.210	1.230	1.117	1.308	1.268	**1.337**	-9.540
gina	**9.965**	0.677	1.283	1.651	2.188	3.567	-15.811
blood-transfusion	-0.665	**0.846**	-11.600	**0.846**	**0.846**	**0.846**	-2.150
disclosure_z	**7.270**	0.254	-3.710	0.254	0.254	-2.259	0.254
wilt	**-3.685**	-12.145	-6.837	-12.145	-12.145	-14.004	-14.657
stock	**1.523**	-3.614	-6.084	-3.379	-6.084	-0.670	1.505
ar4	2.704	3.438	-10.190	**4.213**	0.999	-4.535	-4.936
fri_c4_250_100	**43.762**	17.724	10.730	10.351	12.254	14.855	-15.203
clean2	1.766	6.351	**8.609**	7.671	7.928	4.096	-8.298
philippine	**36.231**	-0.597	-5.838	2.114	2.618	1.461	-2.119
Count	9	2	1	4	2	3	2

KNN – AUC (%)

Dataset	SFS	Gini	ReliefF	JMI	CMIM	mRMR	SPEC
confidence	**-2.576**	**-2.576**	**-2.576**	-8.537	-8.537	**-2.576**	**-2.576**
fri_c3_250_10	27.609	**27.61**	**27.61**	**27.61**	14.445	14.445	-43.587
page-blocks	**0.978**	-0.882	-5.647	-4.414	-4.414	-0.879	-0.981
delta_elevators	**0.069**	0.067	0.067	0.067	0.067	0.067	-33.69
synthetic_control	**0**	**0**	-0.633	**0**	**0**	**0**	**0**
isolet	**0.039**	0.005	**-0.039**	0.038	-0.83	-0.021	-9.633
mfeat-zernike	**-0.2**	-1.018	-0.51	-0.601	-1.084	-0.338	-20.553
gina	**6.335**	0.06	1.479	0.358	2.036	2.518	-21.406
blood-transfusion	**-2.311**	-6.112	-3.389	-6.112	-6.112	-6.112	-2.55
disclosure_z	5.763	5.467	0.881	5.467	5.467	**5.764**	5.467
wilt	1.982	-37.58	**1.983**	-37.58	-37.58	-37.081	-37.592
stock	**0.451**	-0.652	-0.282	0.222	-0.282	-0.711	-3.843
ar4	**18.142**	0.715	0.162	4.526	2.997	0.192	10.169
fri_c4_250_100	**49.295**	40.628	23.774	30.759	13.594	36.088	-5.506
clean2	**0.455**	0.192	-0.005	0.285	0.173	-0.748	-5.841
philippine	**40.394**	21.846	-0.021	5.743	5.934	1.472	-1.662
Count	13	3	4	2	1	3	2

Objectives

1. Study distributed feature selection methods for:
    a. comparison of several search algorithms of wrapper feature selection through evaluation of the quality of final feature subset selected by each method and
    b. economic cost analysis of parallelism of FS methods using cloud-computing-assisted learning methods
2. Optimize wrapper feature selection methods by adopting known distribution optimization techniques such as distributed frameworks with parallel computing, parallel programming methods, and several load partitioning and communication methods for distributed feature selection
3. Propose a novel adaptive feature selection method for data streams using online machine learning
4. Optimize the distribution and parallelism of feature engineering for Big Data
    a. Analyze the scalability of feature engineering methods