Distribution and Replication for Feature Selection

(ESR 2.2)

Uchechukwu Njoku

1st DEDS Winter school

Athens, Greece

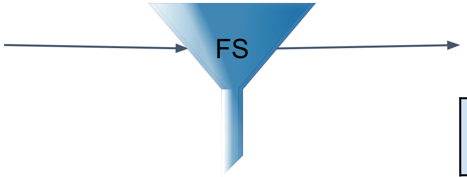
5th April, 2022

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** [1]

$\{F_1, F_2, ..., F_n\}$



$\{F_1, F_2, ..., F_m\}$

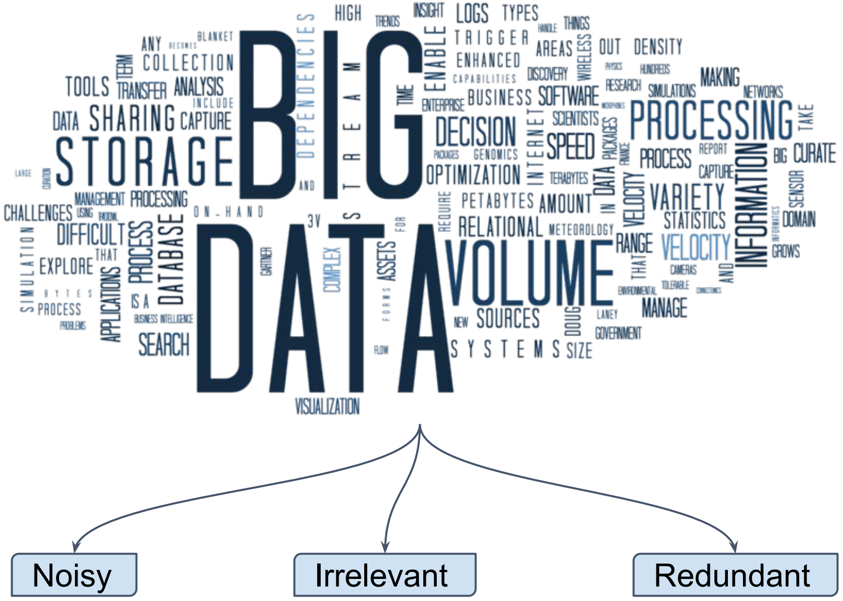
F_1	F_2	F_3	…	F_m	…	F_n
c_{11}	c_{12}	c_{13}	…	c_{1m}	…	c_{1n}
c_{21}	c_{22}	c_{23}	…	c_{2m}	…	c_{2n}
…	…	…	…	…	…	…
c_{j1}	c_{j2}	c_{j3}	…	c_{jm}	…	c_{jn}

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?



Noisy

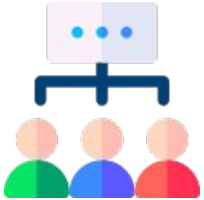
Irrelevant

Redundant

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

Applications

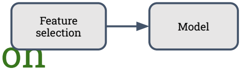
- ○ Text classification
- ○ DNA microarray analysis (6000 → 60,000) [2]
- ○ Image classification
- ○ Face recognition
- ○ Telecommunications



FS Classification

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

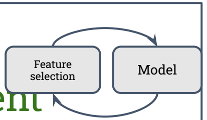
Filter methods



- ○ Fast execution
- ○ Good generalization
- ○ Robust to overfitting
- ○ Possible redundancy
- ○ Model independent
- ○ Non ‘optimal’ selection

E.g: Gini, ReliefF, MRMR, CFS

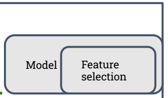
Wrapper methods



- ○ 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

An Example

- Synthetic_control
- 61 features
- 600 instances
- 30 features
- Decision tree

Filter (Gini) → 0.935

Wrapper (SFS) → 0.971

Embedded → 0.89

Baseline → 0.97

Limitations

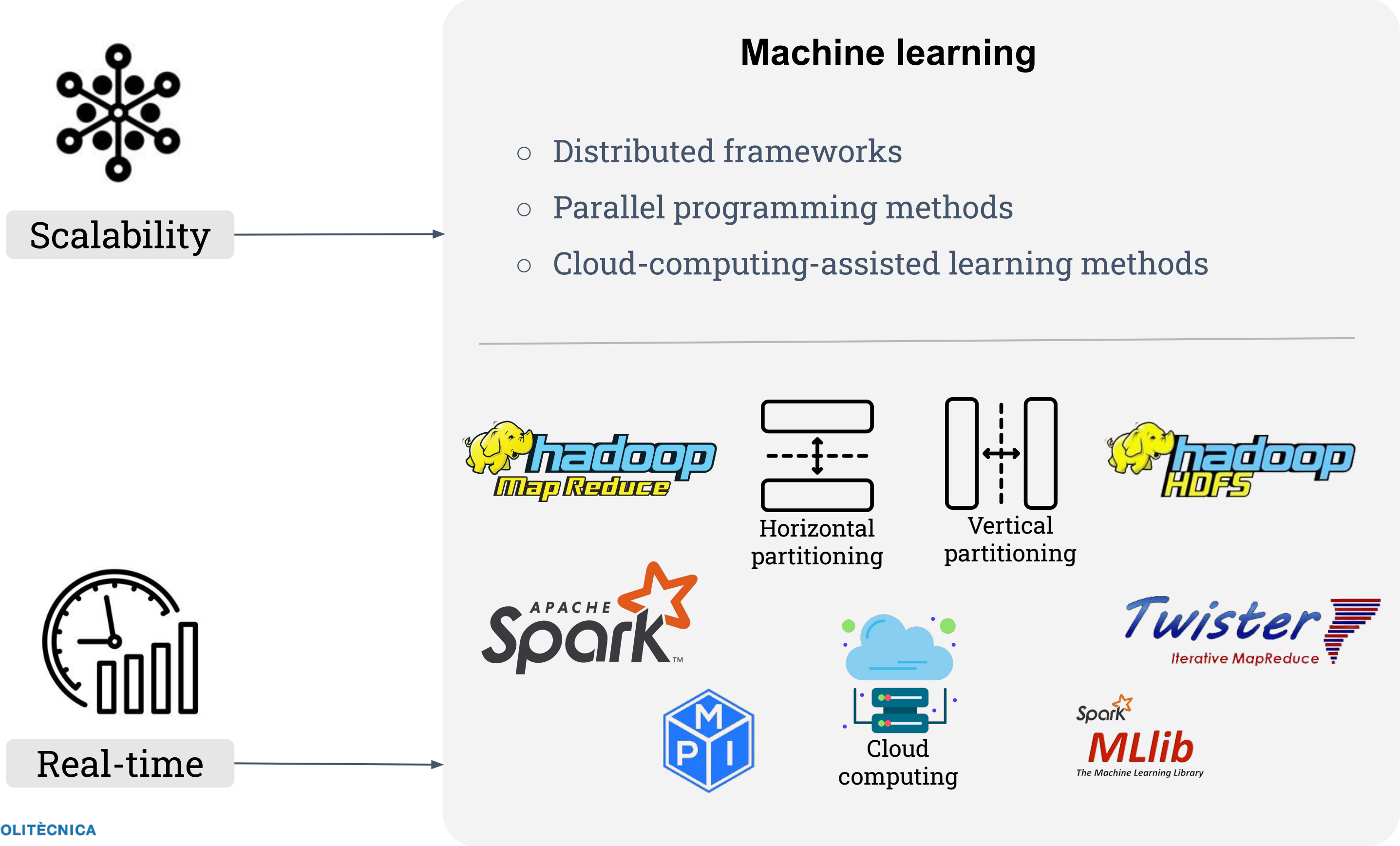
○ Existing methods were developed for small-medium dataset sizes

○ Assuming that the entire dataset fits in the main memory

Scalability

Real-time

Techniques



Scalability

Real-time

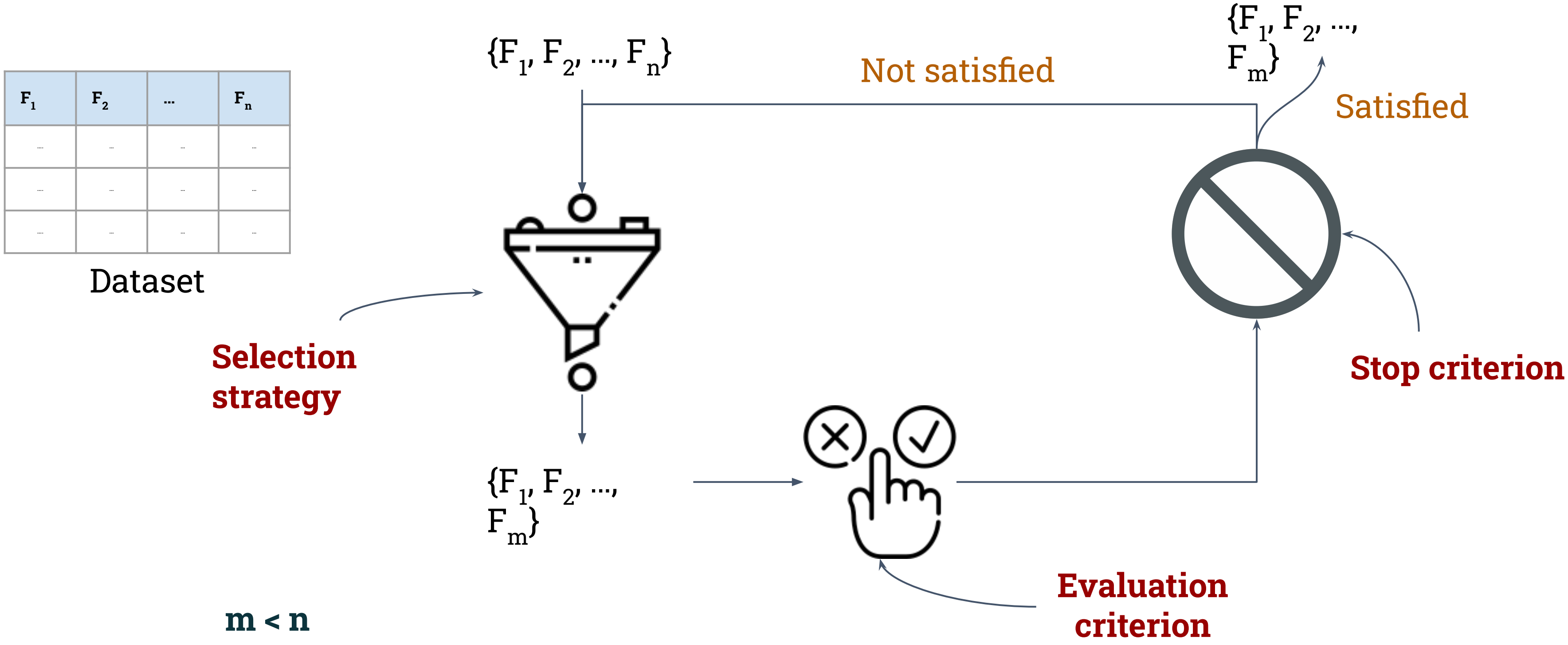
Machine learning

- Distributed frameworks
- Parallel programming methods
- Cloud-computing-assisted learning methods

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

Wrapper Feature Selection



F_1	F_2	...	F_n
...	...	...	...
...	...	...	...
...	...	...	...

Dataset

$\{F_1, F_2, ..., F_n\}$

Not satisfied

$\{F_1, F_2, ..., F_m\}$

Satisfied

Selection strategy

$\{F_1, F_2, ..., F_m\}$

Stop criterion

Evaluation criterion

$m < n$

Selection Strategies

Exponential

E.g: Exhaustive search

- Considers all possible feature subset (2^n - 1)
- Guarantees to find optimal solution

- Exponential complexity (Impractical)
- Difficult to run even for moderate feature size

Population Based

E.g: Genetic algorithms

- Good tradeoff between computational cost and quality of solution

- Several parameters (affects outcome)
- Initialization strategy (affects outcome)
- Slow convergence

Sequential

E.g: Sequential forward selection

- Quick to converge

- No guarantee of finding optimal solution

Evaluation criterion

Performance measure of learning algorithm

- Accuracy
- AUC score
- Precision
- Recall
- F1
- Jaccard
- rand_score

Stop criteria

- Set performance threshold
- Number of selected features
- Number of iterations/generations
- Number of consecutive rounds without improvement

Related Works

Year	Paper	Approach
2013	A Distributed Wrapper Approach for Feature Selection [3]	Selection strategy: **Sequential Forward Selection** Equal **vertical partitioning** guided by information gain **Result** → Shortened execution time with maintained accuracy
2015	Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach [4]	Selection strategy: **Population based (CHC)** Horizontal partitioning in HDFS, Mapreduce and MLLib (KNN) in Spark for selection and evaluation. Feature weight for aggregation **Result** → reduce adequately the number of features of large datasets, smaller storage, faster computation, and easier classification
2018	A Preliminary Study of the Feasibility of Global Evolutionary Feature Selection for Big Datasets under Apache Spark [5]	Selection strategy: **Population based (CHC)** MLLib (Decision Tree) in Spark for evaluation only **Result** → handle very big datasets with a large number of instances and features

Gaps

Experiments show success of these in dealing with the challenge of poor scalability

- **Few works**
- **No tools**
- **Fixed evaluation**

Horizontal partitioning

hadoop Map Reduce

Spark MLlib The Machine Learning Library

APACHE Spark™

Online FS

Why?

- ○ Growing features/instances
- ○ Traditional batch approach is insufficient

Existing solutions

- ○ Grafting [6]
- ○ Online feature selection (OFS) [6]
- ○ Online streaming feature selection (OSFS) [6]
- ○ Feature ranking method + incremental learning algorithm [7]

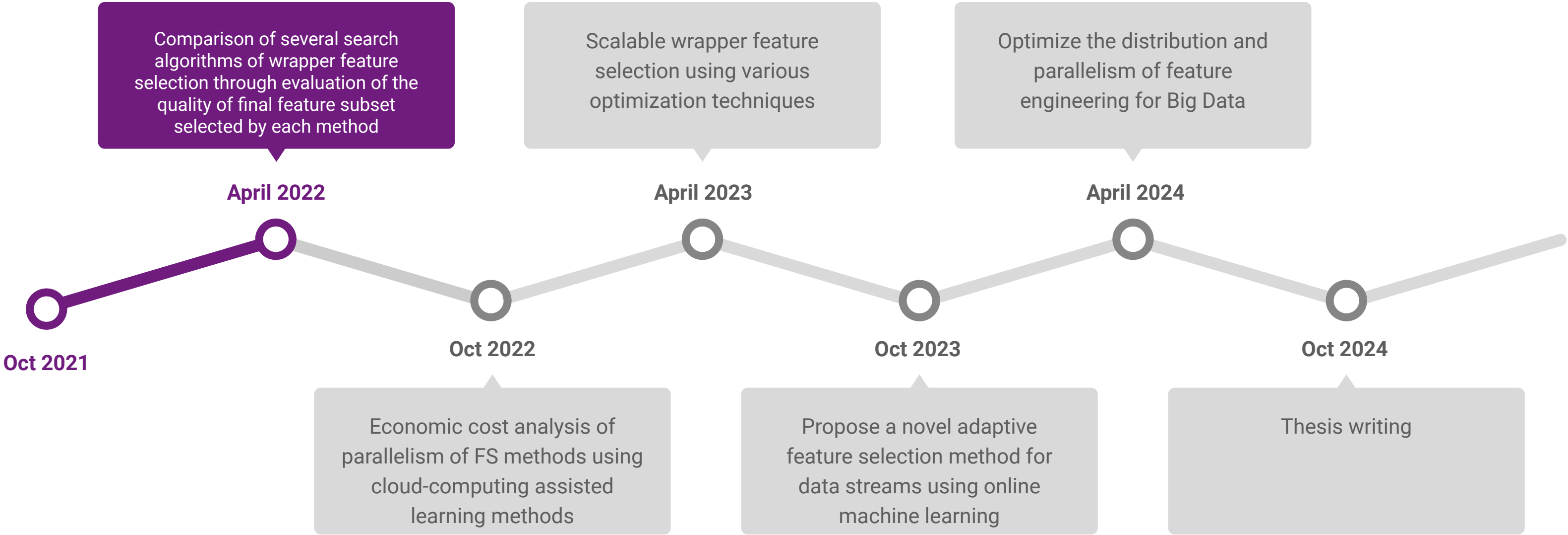
Limitations

- ○ Unvalidated with online ML algorithms
- ○ Limited to feature-based learning algorithms

Objectives

1. Study present 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

Timeline



Oct 2021

April 2022

Comparison of several search algorithms of wrapper feature selection through evaluation of the quality of final feature subset selected by each method

Oct 2022

Economic cost analysis of parallelism of FS methods using cloud-computing assisted learning methods

April 2023

Scalable wrapper feature selection using various optimization techniques

Oct 2023

Propose a novel adaptive feature selection method for data streams using online machine learning

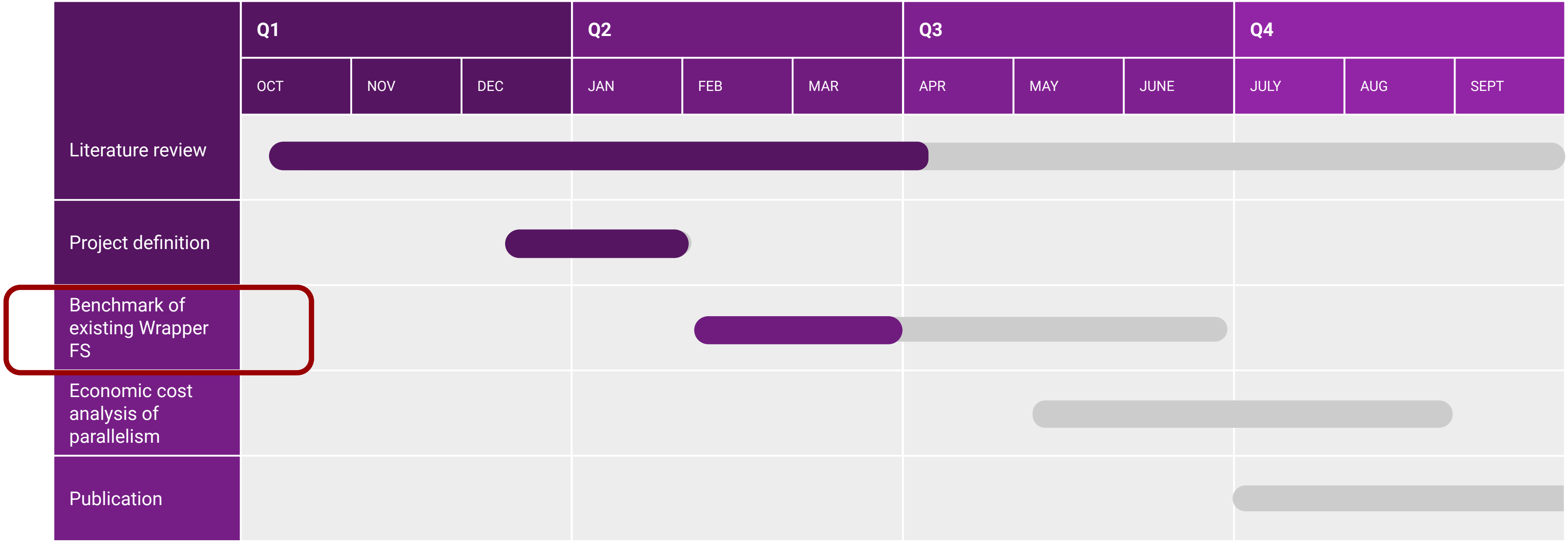
April 2024

Optimize the distribution and parallelism of feature engineering for Big Data

Oct 2024

Thesis writing

Timeline



Q1 Q2 Q3 Q4

OCT NOV DEC JAN FEB MAR APR MAY JUNE JULY AUG SEPT

Literature review

Project definition

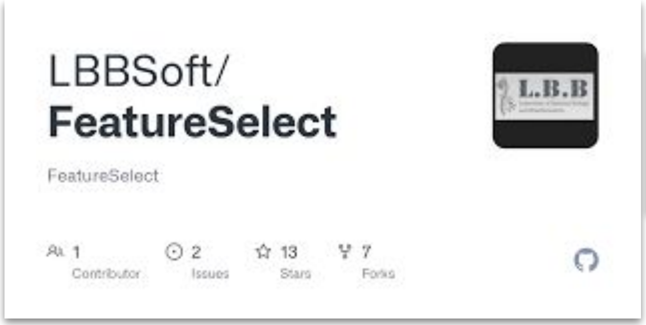
Benchmark of existing Wrapper FS

Economic cost analysis of parallelism

Publication

A survey of Wrapper Feature Selection tools

Tools for wrapper FS



LBBSoft/**FeatureSelect**

FeatureSelect

1 Contributor · 2 Issues · 13 Stars · 7 Forks

jundongl/**scikit-feature**

open-source feature selection repository in python

6 Contributors · 29 Issues · 1k Stars · 407 Forks

Tools: 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

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

References

1. Bolón-Canedo, Verónica, et al. "Feature Selection: From the Past to the Future." Advances in Selected Artificial Intelligence Areas. Springer, Cham, (2022)
2. Bolón-Canedo, Verónica, et al. "A review of microarray datasets and applied feature selection methods." Information sciences 282 (2014):
3. Bolón-Canedo, Verónica, Noelia Sánchez-Marono, and Amparo Alonso-Betanzos. "A distributed wrapper approach for feature selection." ESANN. (2013)
4. Peralta, Daniel, et al. "Evolutionary feature selection for big data classification: A mapreduce approach." Mathematical Problems in Engineering 2015 (2015)
5. Galar, Mikel, et al. "A preliminary study of the feasibility of global evolutionary feature selection for big datasets under Apache Spark." 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, (2018)
6. Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. "Recent advances and emerging challenges of feature selection in the context of big data." Knowledge-based systems 86 (2015)
7. Katakis, Ioannis, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the utility of incremental feature selection for the classification of textual data streams." Panhellenic Conference on Informatics. Springer, Berlin, Heidelberg, (2005)

THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BIYAN
SHUKRIA
TINGKI
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
MEHRBANI
GRAZIE
MAAKE
KOMAPSUMNIDA
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
PALDIES
BOLZÏN
MERCI
SPASSIBO
SHACHALHUYA
NUHUH
CHALTU
YUSPAGARATAM
WABEEJA
MAITEKA
HUI
DHANYABAAD
ANNHA
ATTO
MERSI
SPASIBO
DENKAUJA
NENACHALHYA
UNALCHEESH
GUI
HATUR
EKOJU
SIKOMO
MAKETAI
MINMONCHAR
SAHCO
MERASTAWHY
GAEJTHO
LAH
AGUYJE
FAKAAUE
TAVTAPUCH
MEDAWAGSE
BAIKA