

Towards the Assisted Design of Data Science Pipelines

Sergey Redyuk (TU Berlin, sergey.redyuk@tu-berlin.de)

Nov 24, 2022





End-to-End Management of Experimental Data Science on Biomedical Molecular Data

Sergey Redyuk, 4th year (2018-2023)

Supervisors:

Prof. Volker Markl (TU Berlin)

Prof. Uwe Ohler (MDC)

Dr. Zoi Kaoudi (TU Berlin)

Prof. Sebastian Schelter (University of Amsterdam)

<https://sergred.github.io/>



Goal Assist end-users in performing data science tasks that are too complex, time-consuming, or overwhelming (*automate & facilitate*)

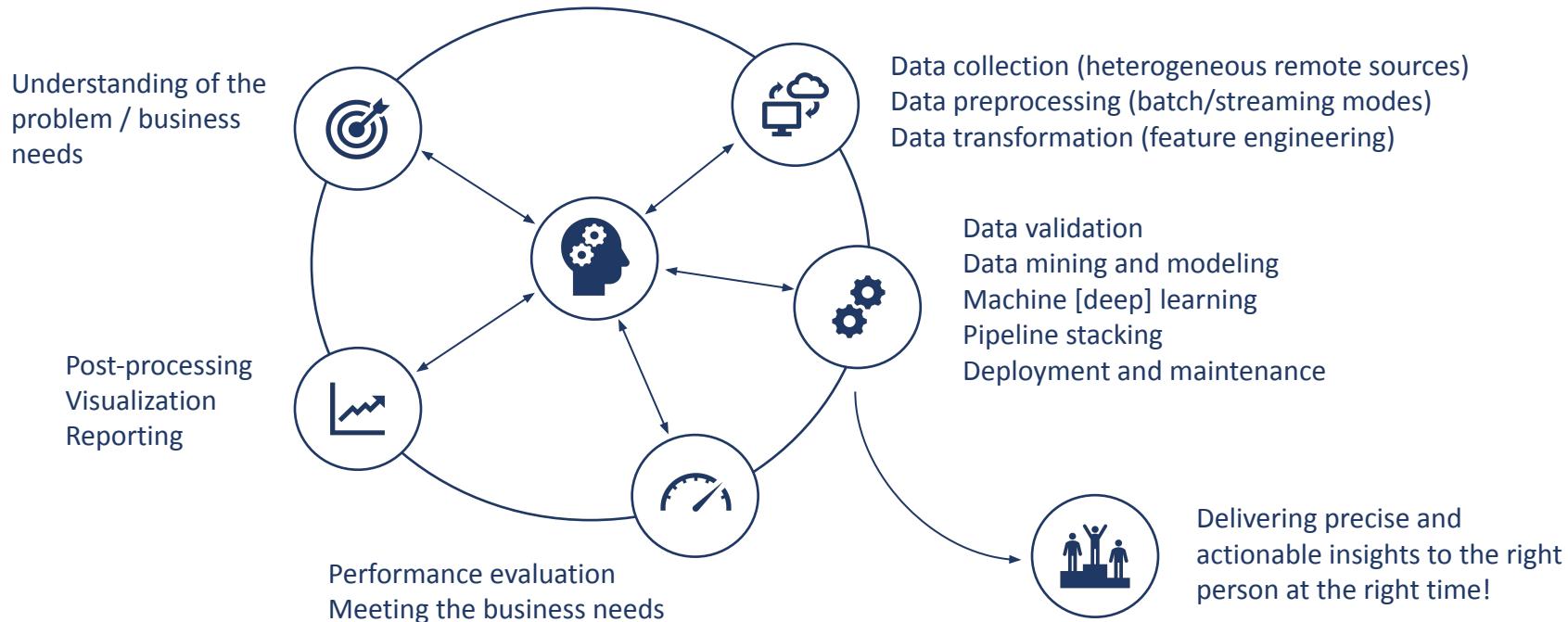
Examples

- **Automated Documentation of Data Science Experiments** - how to automatically detect and track relevant information and digital artifacts
- **Unsupervised Data Quality Validation** - how to automatically detect data quality issues without continuous manual inspection of data pipelines
- **Validating the predictions of Black-Box ML models** - how to demonstrate the effectiveness of Black-Box ML models on previously unseen production data
- **Assisted Design of Data Science Pipelines** - how to help novice-users or domain experts design efficient end-to-end DS pipelines

Selected Publications

- Automated Documentation of End-to-End Experiments in Data Science, PhD Workshop, ICDE'19
- Learning to Validate the Predictions of Black Box Machine Learning Models on Unseen Data, HILDA, SIGMOD'19
- Towards Unsupervised Data Quality Validation on Dynamic Data, ETMLP, EDBT'20
- Automating Data Quality Validation for Dynamic Data Ingestion, EDBT'21
- DORIAN in Action: Assisted Design of Data Science Pipelines, VLDB'22

Data Science Processes

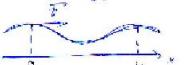


A Lab Notebook is ... *

- Complete **record of procedures**, reagents, **data**, and **thoughts** to pass on to other researchers
- **Explanation** of why experiments were initiated, how they were performed, and the results
- **Main source for reproducibility of experiments**
- **Legal document** to prove patents and defend your data against accusations of fraud
- **Scientific legacy** in the lab

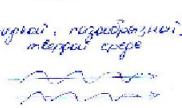
Блокнот - это **важное утверждение** о том, что вы делали.

Блокнот - это **запись** логичных **степеней** (стадий) в **специальном** **формате**.



$$f(x, \varepsilon) = \begin{cases} f(x - \frac{\varepsilon}{2}) - \varepsilon & \text{если } x < 0 \\ f(x) + \frac{\varepsilon}{2} & \text{если } 0 < x < 1 \\ f(x + \frac{\varepsilon}{2}) - \varepsilon & \text{если } x > 1 \end{cases}$$

Приближение
(помимо конечного ядра)
непрерывных производных
или более



Приближение
(помимо конечного ядра)
непрерывных производных
или непрерывных изломов



Блокнот - это:

$$\begin{aligned} g &= f(x - \frac{\varepsilon}{2}) \\ g &= f - \varepsilon/2 \end{aligned}$$

$$\frac{\partial g}{\partial x} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial x} = f' = g'$$

$$\frac{\partial g}{\partial x} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial x} = f' - \frac{\varepsilon}{2} = -\frac{\varepsilon}{2}$$

$$\begin{aligned} \frac{\partial^2 g}{\partial x^2} &= \frac{\partial f}{\partial x} \cdot \frac{\partial^2 x}{\partial x^2} + (-\frac{\varepsilon}{2}) \frac{\partial^2 f}{\partial x^2} = \frac{\varepsilon}{2} - \frac{\varepsilon^2}{4} = \frac{\varepsilon}{2} - \frac{\varepsilon^2}{4} \\ \frac{\partial^2 g}{\partial x^2} &= \frac{1}{2} \frac{\varepsilon^2}{2} - \text{непрерывные производные ядра} \end{aligned}$$

$\frac{\partial g}{\partial x} + \varepsilon/2 = \text{непрерывные производные ядра, планирующие ядро}$

$\frac{\partial^2 g}{\partial x^2} = \text{непрерывные производные ядра}$



излом ядро непрерывные ядро
излом ядро

$E = \frac{f_2}{f_1} - \text{доля производств}$

$E > 0 - \text{производство}$

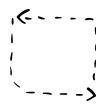
$E < 0 - \text{избыток}$

* Keeping a Lab Notebook, NIH, Office of Intramural Training and Education
[[https://www.training.nih.gov/assets/Lab_Notebook_508_\(new\).pdf](https://www.training.nih.gov/assets/Lab_Notebook_508_(new).pdf)]

Automated Documentation of Data Science Experiments

80% of workload – solving technical problems

- Abundance of tools and frameworks – “glue” code, “smells”
- Multi-tenant environment
- Lack of systematic holistic approaches
- Try-Fail-Learn-Iterate paradigm



Reproducibility as the core of the scientific method is at stake

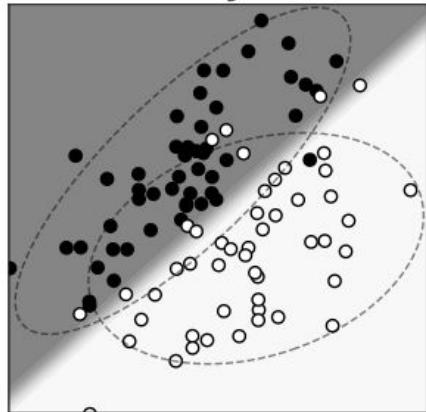


<https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1740-9713.2015.00827.x>

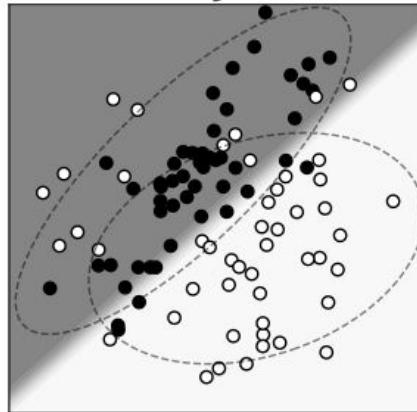
<https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html>

Validating the predictions of Black-Box ML models

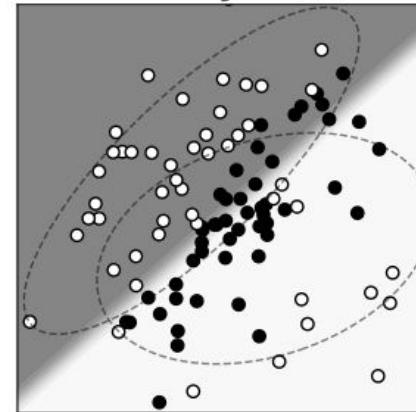
i.i.d. target data:
accuracy = 0.9



25% corruption:
accuracy = 0.82



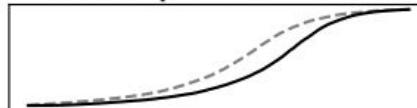
75% corruption:
accuracy = 0.56



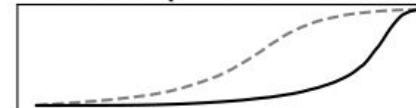
CDF of positive
class predictions



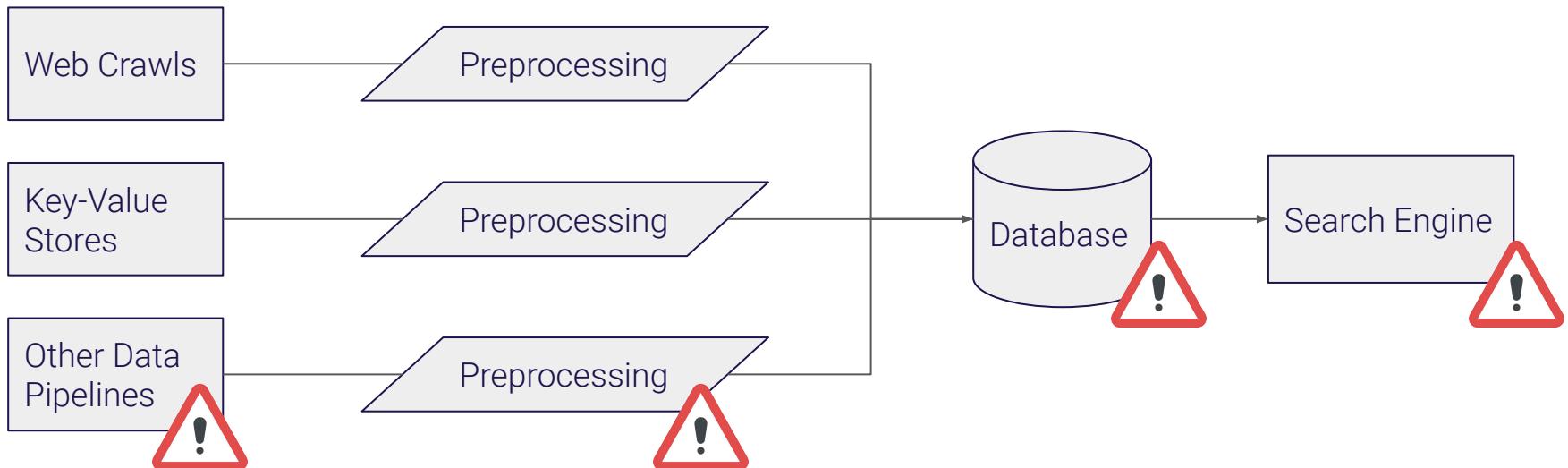
CDF of positive
class predictions



CDF of positive
class predictions



Scenario I. Retail Company



Overall Challenges

Assistance required

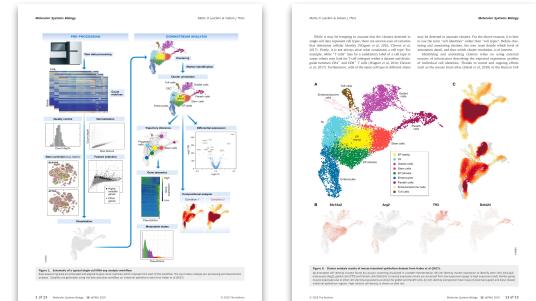
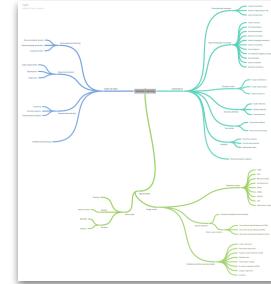
- to bring down costs
- to increase scalability of data ingestion, experimentation, pipeline design, etc.
- to reduce the time domain experts and engineers have to spend on fixing DQ issues, validating the models, aka 'Janitor Work'
- to adapt for non-expert users

Automation is not always possible

- user input needed

Assisted Design of DS Pipelines

- Design of DS pipelines might be **overwhelming** for **domain experts** and novice users
- Even for ML experts, **hard to keep up** with new development
- Assisting tools are **bound** to a particular context
 - Supported DS tasks
 - Supported DS operators
 - Supported evaluation processes
- What if... application domains **surpass** this context



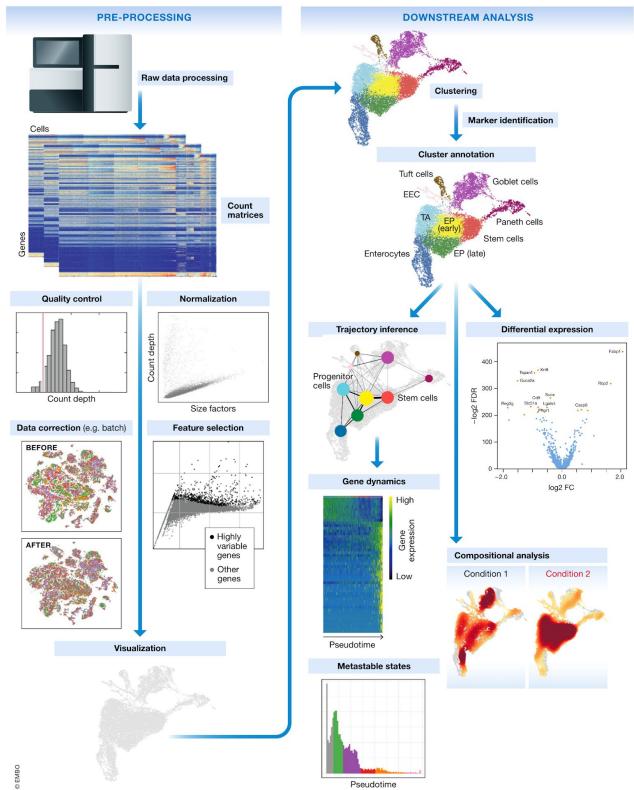


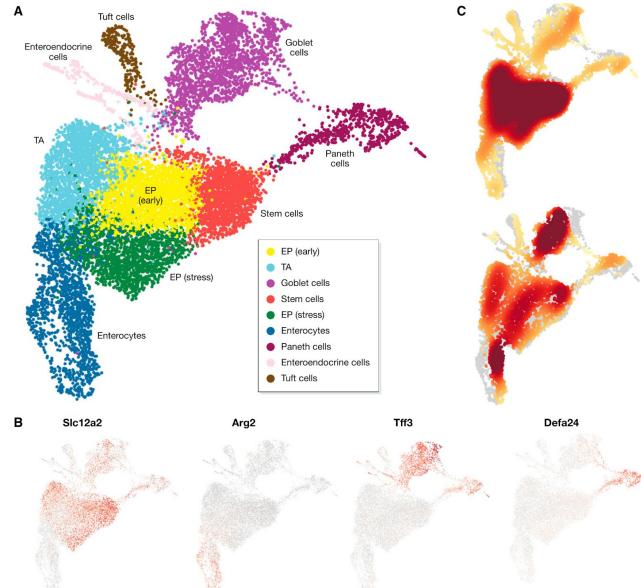
Figure 1. Schematic of a typical single-cell RNA-seq analysis workflow.

Raw sequencing data are processed and aligned to give count matrices, which represent the start of the workflow. The count data undergo pre-processing and downstream analysis. Subplots are generated using the best-practices workflow on intestinal epithelium data from Haber *et al.* (2017).

While it may be tempting to assume that the clusters detected in single-cell data represent cell types, there are several axes of variation that determine cellular identity (Wagner *et al.* 2016; Clevers *et al.* 2017). Firstly, it is not always clear what constitutes a cell type. For example, while “T cells” may be a satisfactory label of a cell type to some, others may look for T-cell subtypes within a dataset and distinguish between CD4⁺ and CD8⁺ T cells (Wagner *et al.* 2016; Clevers *et al.* 2017). Furthermore, cells of the same cell type in different states

may be detected in separate clusters. For the above reasons, it is best to use the term “cell identities” rather than “cell types”. Before clustering and annotating clusters, the user must decide which level of annotation detail, and thus which cluster resolution, is of interest.

Identifying and annotating clusters relies on using external sources of information describing the expected expression profiles of individual cell identities. Thanks to recent and ongoing efforts such as the mouse brain atlas (Zeisel *et al.* 2018) or the Human Cell

Figure 6. Cluster analysis results of mouse intestinal epithelium dataset from Haber *et al.* (2017).

(A) Annotated cell-identity clusters found by Louvain clustering visualized in a tSNE representation. (B) Cell-identity marker expression to identify stem cells (*Slc12a2*), enterocytes (*Arg2*), goblet cells (*Tff3*) and Paneth cells (*Defa24*). Corrected expression levels are visualized from low expression (grey) to high expression (red). Marker genes may be expressed also in other cell-identity populations as shown for goblet and Paneth cells. (C) Cell-identity composition heatmaps of proximal (upper) and distal (lower) intestinal epithelium regions. High relative cell density is shown as dark red.

Auto-Sklearn 2.0: The Next Generation

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer and Frank Hutter

JMLR: Workshop and Conference Proceedings 64:66–74, 2016

ICML 2016 AutoML Workshop

TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning

Randal S. Olson OLSONRAN@UPENN.EDU and Jason H. Moore JHMOORE@UPENN.EDU

Journal of Machine Learning Research 18 (2017) 1–5

Submitted 5/16; Revised 11/16; Published 3/17

Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA

Lars Kotthoff

LARKOT@CS.UBC.CA

Chris Thornton

CWTHORNT@CS.UBC.CA

Holger H. Hoos

HOOS@CS.UBC.CA

Frank Hutter

FH@CS.UNI-FREIBURG.DE

Kevin Leyton-Brown

KEVINL@CS.UBC.CA

Department of Computer Science

University of British Columbia

2366 Main Mall, Vancouver, B.C. V6T 1Z4 Canada

Editor: Geoff Holmes

Abstract

WEKA is a widely used, open-source machine learning platform. Due to its intuitive interface, it is particularly popular with novice users. However, such users often find it hard to identify the best approach for their particular dataset among the many available. We describe the new version of *Auto-WEKA*, a system designed to help such users by automatically searching through the joint space of WEKA’s learning algorithms and their respective hyperparameter settings to maximize performance using a state-of-the-art Bayesian optimization method. Our new package is tightly integrated with WEKA, making it just as accessible to end users as any other learning algorithm.

Keywords: Hyperparameter Optimization, Model Selection, Feature Selection

1. The Principle Behind Auto-WEKA

The WEKA machine learning software (Hall et al., 2009) puts state-of-the-art machine learning into the hands of everyone. However, such users typically know how to choose among the dozens of machine learning procedures implemented in WEKA, and each procedure’s hyperparameter settings to achieve good performance.

Auto-WEKA addresses this problem by treating all of WEKA as a single, highly parametric machine learning framework, and using Bayesian optimization to find a strong instantiation for a given dataset. Specifically, it considers the combined space of WEKA’s learning algorithms $\mathcal{A} = \{A^{(1)}, \dots, A^{(k)}\}$ and their associated hyperparameter spaces $\Lambda^{(1)}, \dots, \Lambda^{(k)}$ and aims to identify the combination of algorithm $A^{(j)} \in \mathcal{A}$ and hyperparameters $\lambda \in \Lambda^{(j)}$ that minimizes cross-validation loss.

$$A_{\lambda^*}^* \in \underset{A^{(j)} \in \mathcal{A}, \lambda \in \Lambda^{(j)}}{\operatorname{argmin}} \sum_{i=1}^k \mathcal{L}\left(A_{\lambda}^{(j)}, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{test}}^{(i)}\right),$$

Thornton et al. (2013) first introduced Auto-WEKA and empirically demonstrated state-of-the-art performance. Here we describe an improved and more broadly accessible implementation of Auto-WEKA, focusing on usability and software design.

©2017 Lars Kotthoff, Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown.

License: CC-BY 4.0, see <https://creativecommons.org/licenses/by/4.0/>. Attribution requirements are provided at <http://jmlr.org/papers/v18/16-261.html>

Designing KDD-Workflows via HTN-Planning for Intelligent Discovery Assistance

Abstract

a lot of
of oper-
support
large nu-
a diffi-
rectness
executin
hours of
This
above p-
matic w-
decoupled
to speci-
covery i-
(DM) is

1. Int

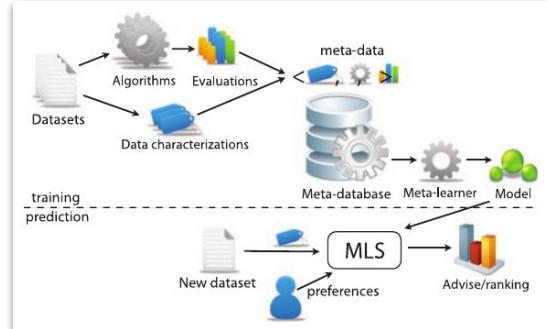
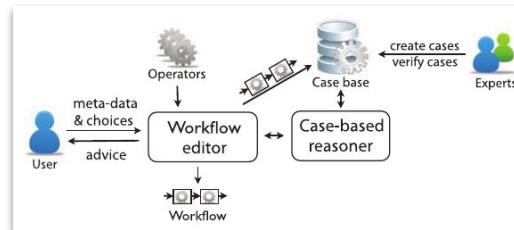
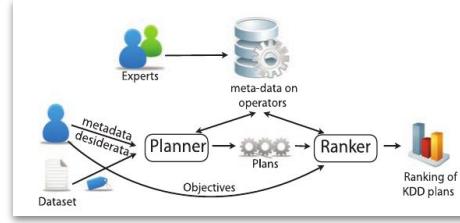
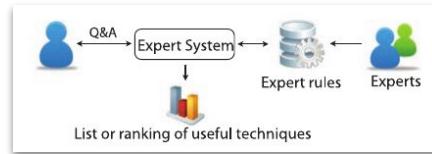
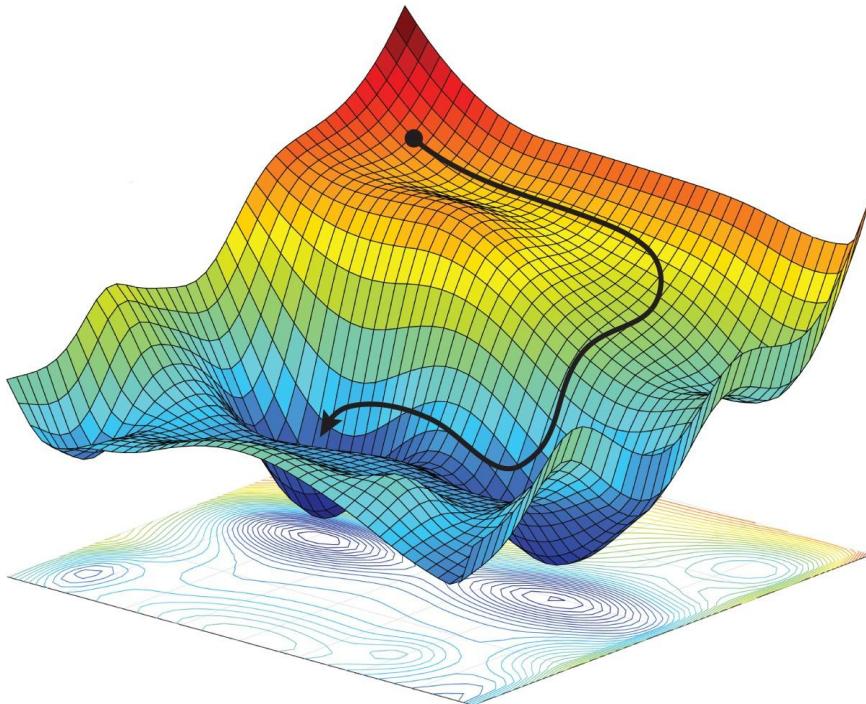
One of
is assist
KDD sy-
open to
graphiti
the wor
operator
become
large.
Howa
tions to
and Rop
erator st
age and
tion fro
tems th
intervene
further a
what on
In ad
workfl

1. Introd

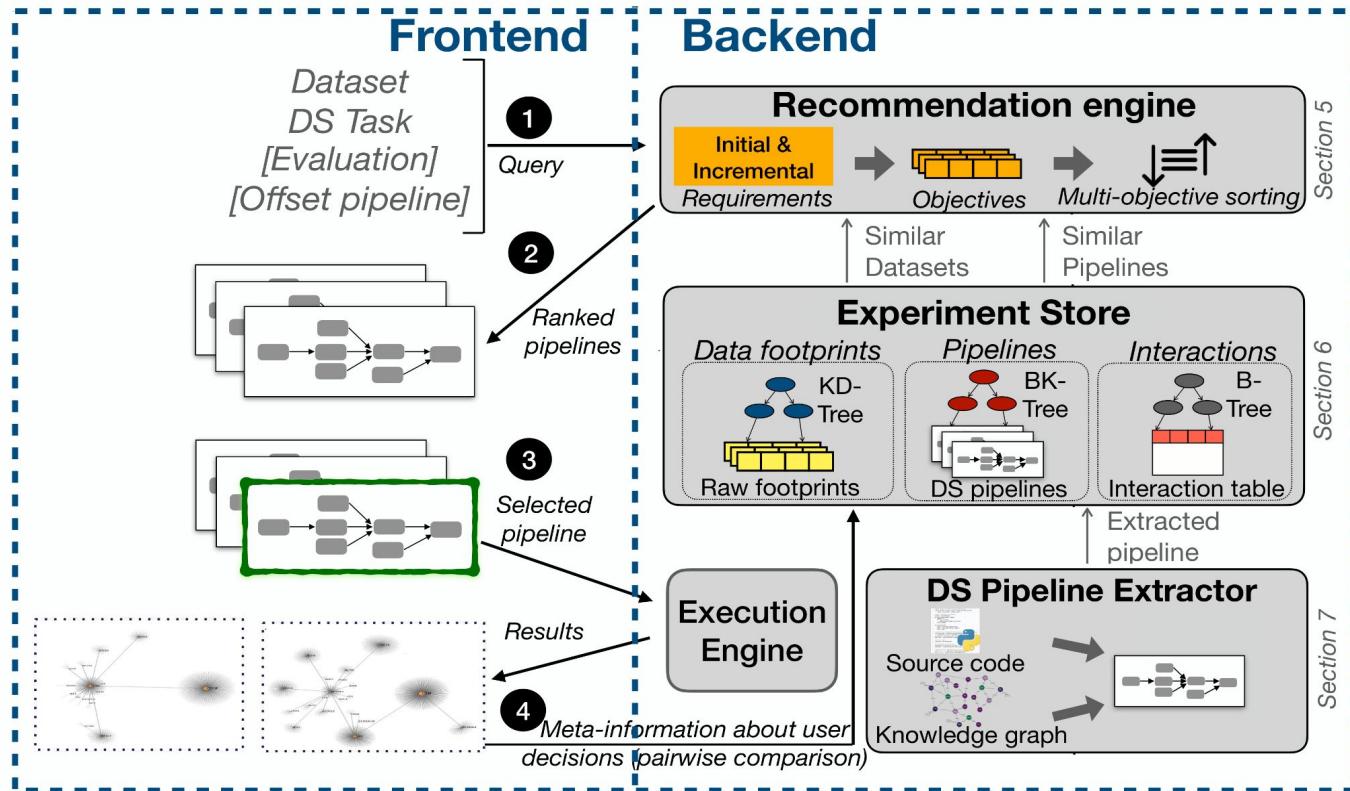
One of
classification
is better
since the
important
ers and pr
classification
using algorit
with a new
with high acc
use? In either
of the NFL T

1. User

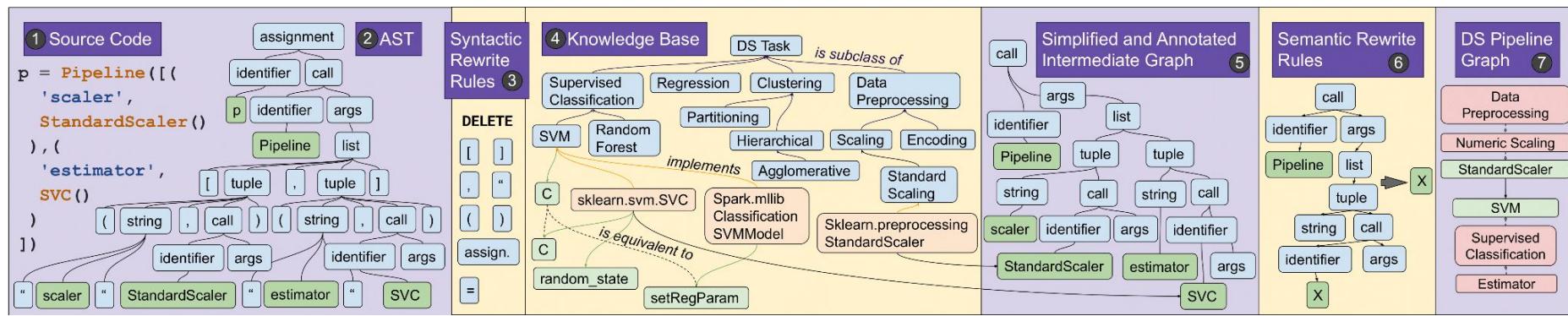
subsys
2. Zersch
3. Rausch
4. flied
5. flied
6. flied
7. flied
8. flied
9. flied
10. flied
11. flied
12. flied
13. flied
14. flied
15. flied
16. flied
17. flied
18. flied
19. flied
20. flied
21. flied
22. flied
23. flied
24. flied
25. flied
26. flied
27. flied
28. flied
29. flied
30. flied
31. flied
32. flied
33. flied
34. flied
35. flied
36. flied
37. flied
38. flied
39. flied
40. flied
41. flied
42. flied
43. flied
44. flied
45. flied
46. flied
47. flied
48. flied
49. flied
50. flied
51. flied
52. flied
53. flied
54. flied
55. flied
56. flied
57. flied
58. flied
59. flied
60. flied
61. flied
62. flied
63. flied
64. flied
65. flied
66. flied
67. flied
68. flied
69. flied
70. flied
71. flied
72. flied
73. flied
74. flied
75. flied
76. flied
77. flied
78. flied
79. flied
80. flied
81. flied
82. flied
83. flied
84. flied
85. flied
86. flied
87. flied
88. flied
89. flied
90. flied
91. flied
92. flied
93. flied
94. flied
95. flied
96. flied
97. flied
98. flied
99. flied
100. flied
101. flied
102. flied
103. flied
104. flied
105. flied
106. flied
107. flied
108. flied
109. flied
110. flied
111. flied
112. flied
113. flied
114. flied
115. flied
116. flied
117. flied
118. flied
119. flied
120. flied
121. flied
122. flied
123. flied
124. flied
125. flied
126. flied
127. flied
128. flied
129. flied
130. flied
131. flied
132. flied
133. flied
134. flied
135. flied
136. flied
137. flied
138. flied
139. flied
140. flied
141. flied
142. flied
143. flied
144. flied
145. flied
146. flied
147. flied
148. flied
149. flied
150. flied
151. flied
152. flied
153. flied
154. flied
155. flied
156. flied
157. flied
158. flied
159. flied
160. flied
161. flied
162. flied
163. flied
164. flied
165. flied
166. flied
167. flied
168. flied
169. flied
170. flied
171. flied
172. flied
173. flied
174. flied
175. flied
176. flied
177. flied
178. flied
179. flied
180. flied
181. flied
182. flied
183. flied
184. flied
185. flied
186. flied
187. flied
188. flied
189. flied
190. flied
191. flied
192. flied
193. flied
194. flied
195. flied
196. flied
197. flied
198. flied
199. flied
200. flied
201. flied
202. flied
203. flied
204. flied
205. flied
206. flied
207. flied
208. flied
209. flied
210. flied
211. flied
212. flied
213. flied
214. flied
215. flied
216. flied
217. flied
218. flied
219. flied
220. flied
221. flied
222. flied
223. flied
224. flied
225. flied
226. flied
227. flied
228. flied
229. flied
230. flied
231. flied
232. flied
233. flied
234. flied
235. flied
236. flied
237. flied
238. flied
239. flied
240. flied
241. flied
242. flied
243. flied
244. flied
245. flied
246. flied
247. flied
248. flied
249. flied
250. flied
251. flied
252. flied
253. flied
254. flied
255. flied
256. flied
257. flied
258. flied
259. flied
260. flied
261. flied
262. flied
263. flied
264. flied
265. flied
266. flied
267. flied
268. flied
269. flied
270. flied
271. flied
272. flied
273. flied
274. flied
275. flied
276. flied
277. flied
278. flied
279. flied
280. flied
281. flied
282. flied
283. flied
284. flied
285. flied
286. flied
287. flied
288. flied
289. flied
290. flied
291. flied
292. flied
293. flied
294. flied
295. flied
296. flied
297. flied
298. flied
299. flied
300. flied
301. flied
302. flied
303. flied
304. flied
305. flied
306. flied
307. flied
308. flied
309. flied
310. flied
311. flied
312. flied
313. flied
314. flied
315. flied
316. flied
317. flied
318. flied
319. flied
320. flied
321. flied
322. flied
323. flied
324. flied
325. flied
326. flied
327. flied
328. flied
329. flied
330. flied
331. flied
332. flied
333. flied
334. flied
335. flied
336. flied
337. flied
338. flied
339. flied
340. flied
341. flied
342. flied
343. flied
344. flied
345. flied
346. flied
347. flied
348. flied
349. flied
350. flied
351. flied
352. flied
353. flied
354. flied
355. flied
356. flied
357. flied
358. flied
359. flied
360. flied
361. flied
362. flied
363. flied
364. flied
365. flied
366. flied
367. flied
368. flied
369. flied
370. flied
371. flied
372. flied
373. flied
374. flied
375. flied
376. flied
377. flied
378. flied
379. flied
380. flied
381. flied
382. flied
383. flied
384. flied
385. flied
386. flied
387. flied
388. flied
389. flied
390. flied
391. flied
392. flied
393. flied
394. flied
395. flied
396. flied
397. flied
398. flied
399. flied
400. flied
401. flied
402. flied
403. flied
404. flied
405. flied
406. flied
407. flied
408. flied
409. flied
410. flied
411. flied
412. flied
413. flied
414. flied
415. flied
416. flied
417. flied
418. flied
419. flied
420. flied
421. flied
422. flied
423. flied
424. flied
425. flied
426. flied
427. flied
428. flied
429. flied
430. flied
431. flied
432. flied
433. flied
434. flied
435. flied
436. flied
437. flied
438. flied
439. flied
440. flied
441. flied
442. flied
443. flied
444. flied
445. flied
446. flied
447. flied
448. flied
449. flied
450. flied
451. flied
452. flied
453. flied
454. flied
455. flied
456. flied
457. flied
458. flied
459. flied
460. flied
461. flied
462. flied
463. flied
464. flied
465. flied
466. flied
467. flied
468. flied
469. flied
470. flied
471. flied
472. flied
473. flied
474. flied
475. flied
476. flied
477. flied
478. flied
479. flied
480. flied
481. flied
482. flied
483. flied
484. flied
485. flied
486. flied
487. flied
488. flied
489. flied
490. flied
491. flied
492. flied
493. flied
494. flied
495. flied
496. flied
497. flied
498. flied
499. flied
500. flied
501. flied
502. flied
503. flied
504. flied
505. flied
506. flied
507. flied
508. flied
509. flied
510. flied
511. flied
512. flied
513. flied
514. flied
515. flied
516. flied
517. flied
518. flied
519. flied
520. flied
521. flied
522. flied
523. flied
524. flied
525. flied
526. flied
527. flied
528. flied
529. flied
530. flied
531. flied
532. flied
533. flied
534. flied
535. flied
536. flied
537. flied
538. flied
539. flied
540. flied
541. flied
542. flied
543. flied
544. flied
545. flied
546. flied
547. flied
548. flied
549. flied
550. flied
551. flied
552. flied
553. flied
554. flied
555. flied
556. flied
557. flied
558. flied
559. flied
560. flied
561. flied
562. flied
563. flied
564. flied
565. flied
566. flied
567. flied
568. flied
569. flied
570. flied
571. flied
572. flied
573. flied
574. flied
575. flied
576. flied
577. flied
578. flied
579. flied
580. flied
581. flied
582. flied
583. flied
584. flied
585. flied
586. flied
587. flied
588. flied
589. flied
590. flied
591. flied
592. flied
593. flied
594. flied
595. flied
596. flied
597. flied
598. flied
599. flied
600. flied
601. flied
602. flied
603. flied
604. flied
605. flied
606. flied
607. flied
608. flied
609. flied
610. flied
611. flied
612. flied
613. flied
614. flied
615. flied
616. flied
617. flied
618. flied
619. flied
620. flied
621. flied
622. flied
623. flied
624. flied
625. flied
626. flied
627. flied
628. flied
629. flied
630. flied
631. flied
632. flied
633. flied
634. flied
635. flied
636. flied
637. flied
638. flied
639. flied
640. flied
641. flied
642. flied
643. flied
644. flied
645. flied
646. flied
647. flied
648. flied
649. flied
650. flied
651. flied
652. flied
653. flied
654. flied
655. flied
656. flied
657. flied
658. flied
659. flied
660. flied
661. flied
662. flied
663. flied
664. flied
665. flied
666. flied
667. flied
668. flied
669. flied
670. flied
671. flied
672. flied
673. flied
674. flied
675. flied
676. flied
677. flied
678. flied
679. flied
680. flied
681. flied
682. flied
683. flied
684. flied
685. flied
686. flied
687. flied
688. flied
689. flied
690. flied
691. flied
692. flied
693. flied
694. flied
695. flied
696. flied
697. flied
698. flied
699. flied
700. flied
701. flied
702. flied
703. flied
704. flied
705. flied
706. flied
707. flied
708. flied
709. flied
710. flied
711. flied
712. flied
713. flied
714. flied
715. flied
716. flied
717. flied
718. flied
719. flied
720. flied
721. flied
722. flied
723. flied
724. flied
725. flied
726. flied
727. flied
728. flied
729. flied
730. flied
731. flied
732. flied
733. flied
734. flied
735. flied
736. flied
737. flied
738. flied
739. flied
740. flied
741. flied
742. flied
743. flied
744. flied
745. flied
746. flied
747. flied
748. flied
749. flied
750. flied
751. flied
752. flied
753. flied
754. flied
755. flied
756. flied
757. flied
758. flied
759. flied
760. flied
761. flied
762. flied
763. flied
764. flied
765. flied
766. flied
767. flied
768. flied
769. flied
770. flied
771. flied
772. flied
773. flied
774. flied
775. flied
776. flied
777. flied
778. flied
779. flied
780. flied
781. flied
782. flied
783. flied
784. flied
785. flied
786. flied
787. flied
788. flied
789. flied
790. flied
791. flied
792. flied
793. flied
794. flied
795. flied
796. flied
797. flied
798. flied
799. flied
800. flied
801. flied
802. flied
803. flied
804. flied
805. flied
806. flied
807. flied
808. flied
809. flied
810. flied
811. flied
812. flied
813. flied
814. flied
815. flied
816. flied
817. flied
818. flied
819. flied
820. flied
821. flied
822. flied
823. flied
824. flied
825. flied
826. flied
827. flied
828. flied
829. flied
830. flied
831. flied
832. flied
833. flied
834. flied
835. flied
836. flied
837. flied
838. flied
839. flied
840. flied
841. flied
842. flied
843. flied
844. flied
845. flied
846. flied
847. flied
848. flied
849. flied
850. flied
851. flied
852. flied
853. flied
854. flied
855. flied
856. flied
857. flied
858. flied
859. flied
860. flied
861. flied
862. flied
863. flied
864. flied
865. flied
866. flied
867. flied
868. flied
869. flied
870. flied
871. flied
872. flied
873. flied
874. flied
875. flied
876. flied
877. flied
878. flied
879. flied
880. flied
881. flied
882. flied
883. flied
884. flied
885. flied
886. flied
887. flied
888. flied
889. flied
890. flied
891. flied
892. flied
893. flied
894. flied
895. flied
896. flied
897. flied
898. flied
899. flied
900. flied
901. flied
902. flied
903. flied
904. flied
905. flied
906. flied
907. flied
908. flied
909. flied
910. flied
911. flied
912. flied
913. flied
914. flied
915. flied
916. flied
917. flied
918. flied
919. flied
920. flied
921. flied
922. flied
923. flied
924. flied
925. flied
926. flied
927. flied
928. flied
929. flied
930. flied
931. flied
932. flied
933. flied
934. flied
935. flied
936. flied
937. flied
938. flied
939. flied
940. flied
941. flied
942. flied
943. flied
944. flied
945. flied
946. flied
947. flied
948. flied
949. flied
950. flied
951. flied
952. flied
953. flied
954. flied
955. flied
956. flied
957. flied
958. flied
959. flied
960. flied
961. flied
962. flied
963. flied
964. flied
965. flied
966. flied
967. flied
968. flied
969. flied
970. flied
971. flied
972. flied
973. flied
974. flied
975. flied
976. flied
977. flied
978. flied
979. flied
980. flied
981. flied
982. flied
983. flied
984. flied
985. flied
986. flied
987. flied
988. flied
989. flied
990. flied
991. flied
992. flied
993. flied
994. flied
995. flied
996. flied
997. flied
998. flied
999. flied
1000. flied
1001. flied
1002. flied
1003. flied
1004. flied
1005. flied
1006. flied
1007. flied
1008. flied
1009. flied
1010. flied
1011. flied
1012. flied
1013. flied
1014. flied
1015. flied
1016. flied
1017. flied
1018. flied
1019. flied
1020. flied
1021. flied
1022. flied
1023. flied
1024. flied
1025. flied
1026. flied
1027. flied
1028. flied
1029. flied
1030. flied
1031. flied
1032. flied
1033. flied
1034. flied
1035. flied
1036. flied
1037. flied
1038. flied
1039. flied
1040. flied
1041. flied
1042. flied
1043. flied
1044. flied
1045. flied
1046. flied
1047. flied
1048. flied
1049. flied
1050. flied
1051. flied
1052. flied
1053. flied
1054. flied
1055. flied
1056. flied
1057. flied
1058. flied
1059. flied
1060. flied
1061. flied
1062. flied
1063. flied
1064. flied
1065. flied
1066. flied
1067. flied
1068. flied
1069. flied
1070. flied
1071. flied
1072. flied
1073. flied
1074. flied
1075. flied
1076. flied
1077. flied
1078. flied
1079. flied
1080. flied
1081. flied
1082. flied
1083. flied
1084. flied
1085. flied
1086. flied
1087. flied
1088. flied
1089. flied
10



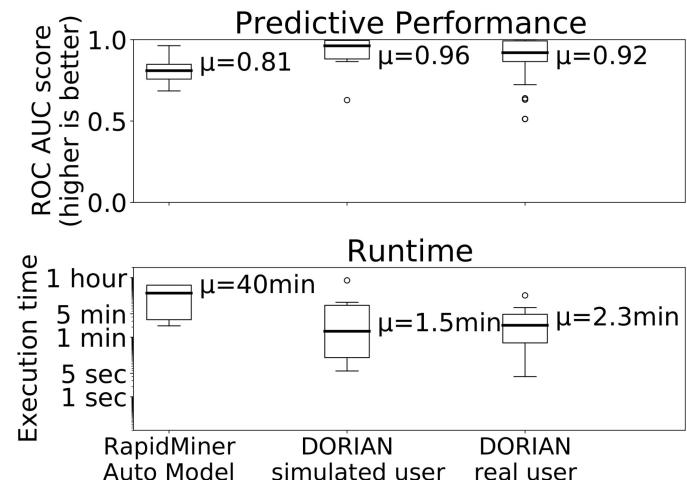
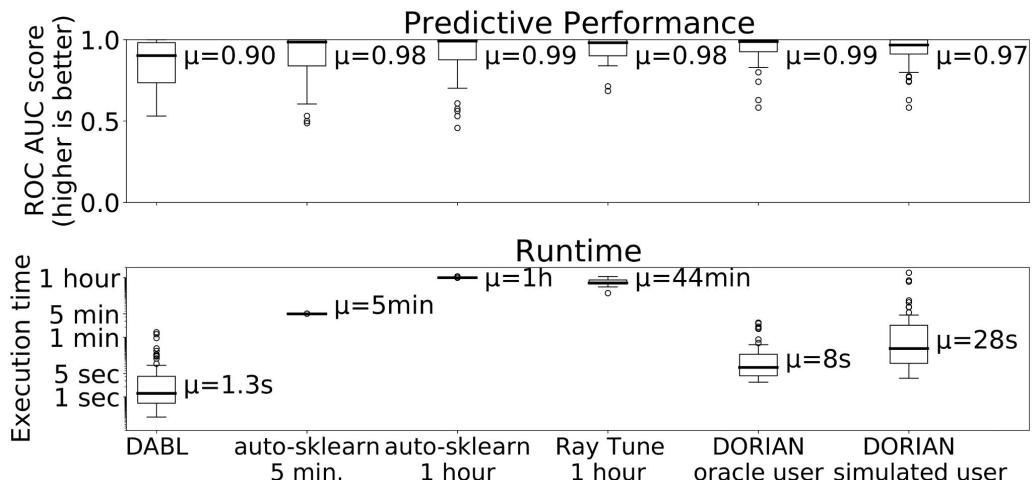
Approach



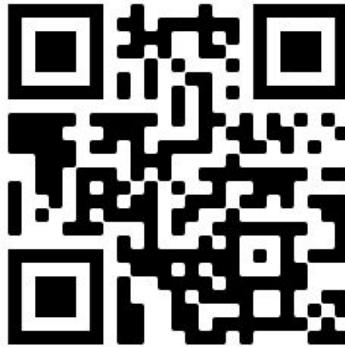
DS Pipeline Extractor



Evaluation, Classification



DORIAN in action: Assisted Design of Data Science Pipelines



- Design of DS pipelines might be overwhelming for domain experts and novice users
- Assisting tools yield limited applicability in a wide range of application domains
- DORIAN is a human-in-the-loop approach for the assisted design of DS pipelines that supports a large and growing set of DS tasks, operators, and arbitrary user-defined evaluation procedures.

Sergey Redyuk (sergey.redyuk@tu-berlin.de), Zoi Kaoudi, Sebastian Schelter, Volker Markl

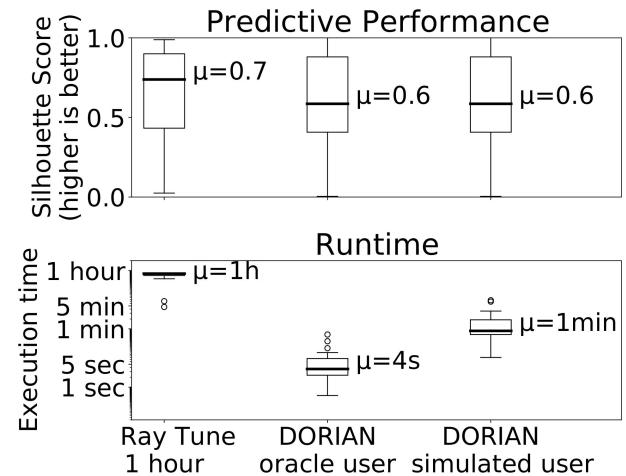
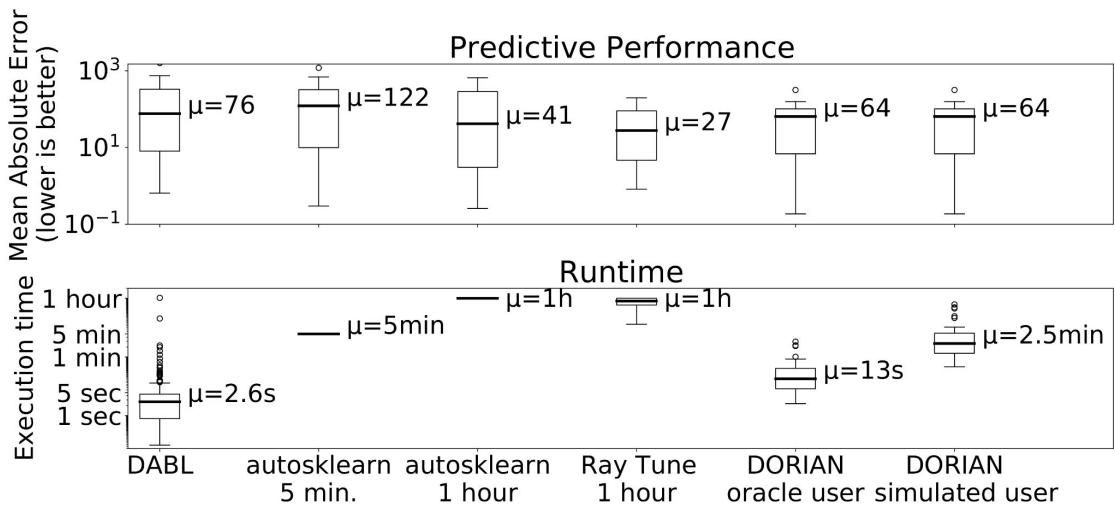


UNIVERSITY
OF AMSTERDAM

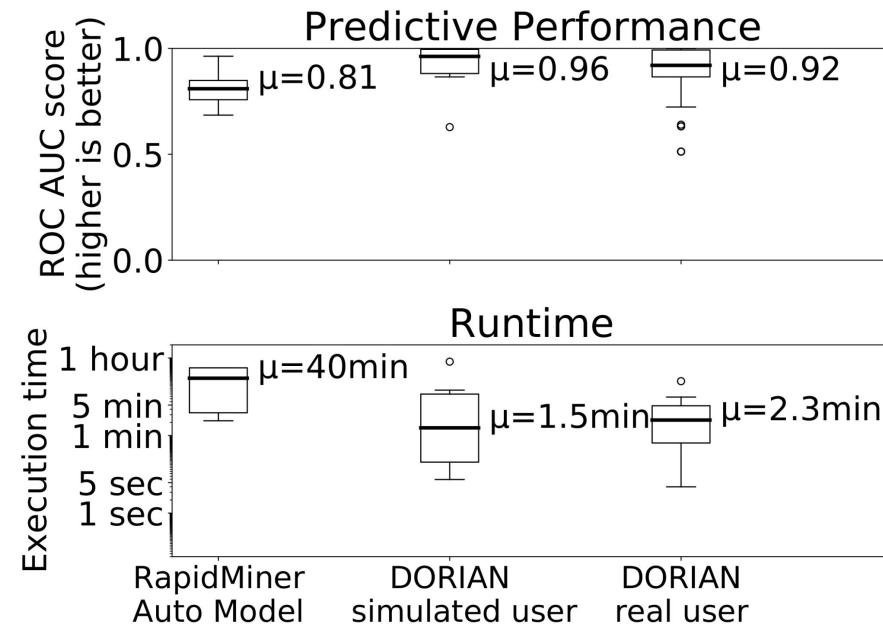
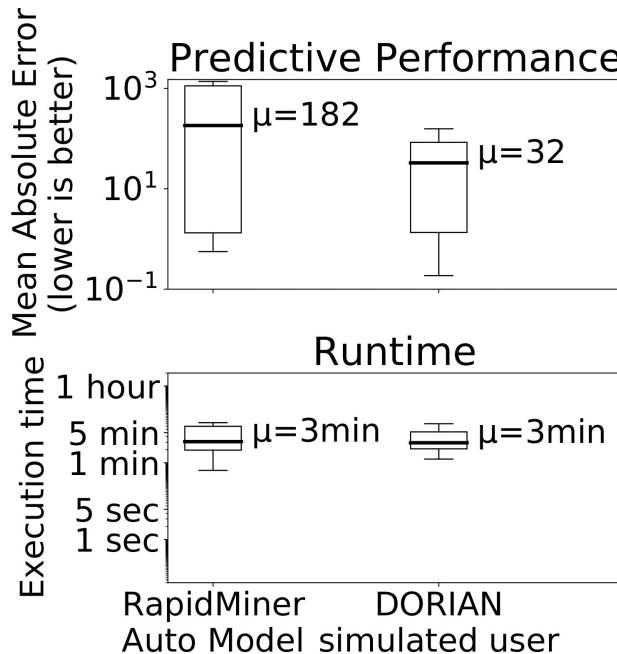
References

- [1] F. Serban, J. Vanschoren, J.-U. Kietz, and A. Bernstein. A survey of intelligent assistants for data analysis. *ACM Computing Surveys (CSUR)*, 45(3):1–35, 2013.
- [2] Z. Shang, E. Zgraggen, B. Buratti, F. Kossmann, P. Eichmann, Y. Chung, C. Binnig, E. Upfal, and T. Kraska. Democratizing data science through interactive curation of ml pipelines. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1171–1188, 2019.
- [3] S. Redyuk, V. Markl, and S. Schelter. Towards unsupervised data quality validation on dynamic data. In *Proceedings of the Workshops of the EDBT/ICDT 2020Joint Conference, Copenhagen, Denmark, March 30, 2020*, volume 2578 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.
- [4] R. S. Olson and J. H. Moore. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning*, pages 66–74. PMLR, 2016.
- [5] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter. Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning*, pages 113–134. Springer, Cham, 2019.
- [6] S. Redyuk, Z. Kaoudi, V. Markl, S. Schelter (2021) Automating Data Quality Validation for Dynamic Data Ingestion. *EDBT'21*, Nicosia, Cyprus
- [7] Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L. and Teal, T.K., 2016. Good Enough Practices in Scientific Computing. *PubMed*
- [8] Polyzotis, N., Roy, S., Whang, S.E. and Zinkevich, M., 2018. Data lifecycle challenges in production machine learning: A survey. *ACM SIGMOD Record*, 47(2), pp.17-28.
- [9] Schelter, S., Biessmann, F., Januschowski, T., Salinas, D., Seufert, S., Szarvas, G., Vartak, M., Madden, S., Miao, H., Deshpande, A. and Zaharia, M., 2018. On Challenges in Machine Learning Model Management. *Data Engineering*, p.5.
- [10] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F. and Dennison, D., 2015. Hidden technical debt in machine learning systems. In *Advances in NIPS* (pp. 2503-2511)

Evaluation, Regression & Clustering



Evaluation, Regression & Clustering (manual)



```

from sklearn.model_selection import train_test_split, GridSearchCV
...
import numpy as np

data_filepath, target = '...', 'class'
data = pd.read_csv(data_filepath)
columns = list(data)

X, y = data[[col for col in columns if col != target]], data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.1)

cat_cols = ['workclass', 'occupation', 'marital_status']
num_cols = ['hours_per_week', 'age']

feature_transformation = ColumnTransformer(transformers=[
    ('cat_features', OneHotEncoder(handle_unknown='ignore'), cat_cols),
    ('scaled_numeric', StandardScaler(), num_cols)
])

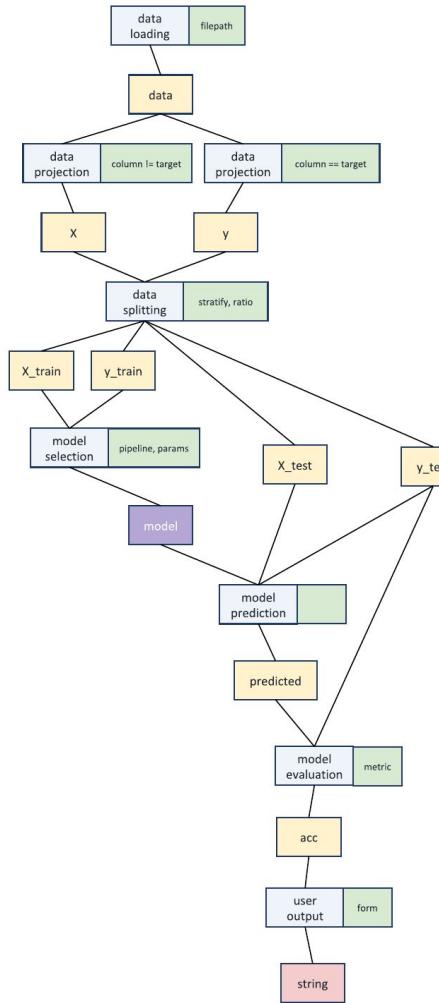
pipeline = Pipeline([
    ('features', feature_transformation),
    ('learner', SGDClassifier(max_iter=1000, tol=1e-3))
])

param_grid = {
    'learner__alpha': [0.0001, 0.001, 0.01, 0.1]
}

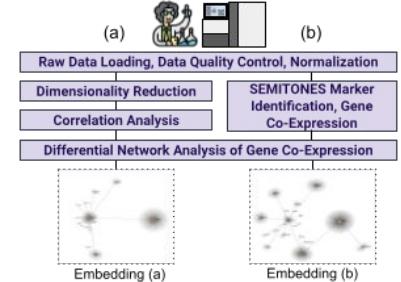
search = GridSearchCV(pipeline, param_grid, cv=5)
model = search.fit(X_train, y_train)

predicted = model.predict(X_test)
acc = accuracy_score(y_test, predicted)
print("TRAIN. accuracy: %.4f" % (acc))

```



Design Decisions



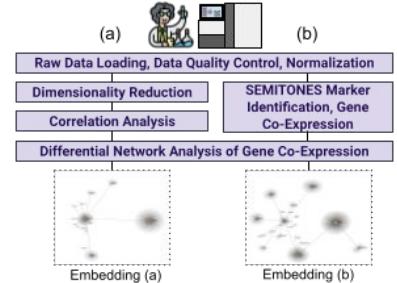
Pipeline as DAG[Operation, ConnectionPoint]

Operation as $f[\text{List}[\text{Input}], \text{List}[\text{Output}]]$, semantically enriched black box

Hyperparameter as special Input

DS ontology to decouple Operation's intent from implementation

Design Decisions [cont.]



Evaluation Process supports arbitrary use cases
(pairwise comparison of candidates)

Interaction Table records user actions and preferences

Ranking Objectives are extendable and take into account two scenarios: where the end-user does or does not specify the offset pipeline (aka Initial recommendations vs Incremental improvements)

Ranking considered a non-dominated sorting problem



git v1.0



Bob

```
# loading the data, local file system
houses = read_csv()
# scatter plot, houses size and price
plot(houses$square, houses$price)
# principal component analysis
components = PCA(houses, houses$price)
# building a linear regression model
model = linear_regression(houses$square,
                           houses$price)
# performance evaluation
error(model.predict(houses$square),
      houses$price) # > 20,000
```



Alice

```
houses = read_csv()
# preparing additional features
center = location((52.5167, 13.3833))
stores = load_store_location()
parks = load_park_location()
# augmenting new features into the data
features = distance(houses$loc,
                     [center, stores, parks])
model = linear_regression(features,
                           houses$price)
error(model.predict(features),
      houses$price) # > 10,000
```



Charlie

```
# loading the data, remote ftp
houses = load_data()
# building the neural network, dense layer
# with 128 filters, 25% dropout, dense
# layer with 64 filters
model = DNN(layers=[

    Dense(128, input=features.shape),
    Dropout(0.25),
    Dense(64),
    Dense(1, activation='linear')
]).fit(houses, houses$price)
error(model.predict(houses),
      houses$price) # > 5,000
```

@data.loading

```
Assign Targets houses
CallFunc read_csv
```

```
Expr CallFunc plot
```

```
Assign Targets components
CallFunc PCA
```

@model.train

```
Assign Targets model
CallFunc linear_regression
```

```
Expr CallFunc error
```

saving the artifacts: models,
charts, performance metrics

```
Assign Targets houses
CallFunc read_csv
```

```
Assign Targets center
CallFunc location -- Args Attr Tuple((52.5167, 13.3833))
```

```
Assign Targets stores
CallFunc load_store_location
```

```
Assign Targets parks
CallFunc load_park_location
```

```
Assign Targets features
CallFunc distance
```

```
Assign Targets model
CallFunc linear_regression
```

```
Expr CallFunc error
```

[background] preparing the predefined mappings 1

2 mapping-based code decomposition



```
Args Attr houses$square
      Attr houses$price
      Attr houses
      Attr houses$price
```

```
Args Attr houses$square
      Attr houses$price
      Attr Attr model predict(houses$square)
      Attr Attr houses$price
```

4

4

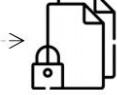
3 multi-user code versioning



```
Args Attr houses$loc
      Attr List([center, stores, parks])
      Attr features
      Attr houses$price
      Attr model predict(houses$square)
      Attr Attr houses$price
```

3

5 saving lineage and metadata



```
Targets houses
CallFunc load_data
```

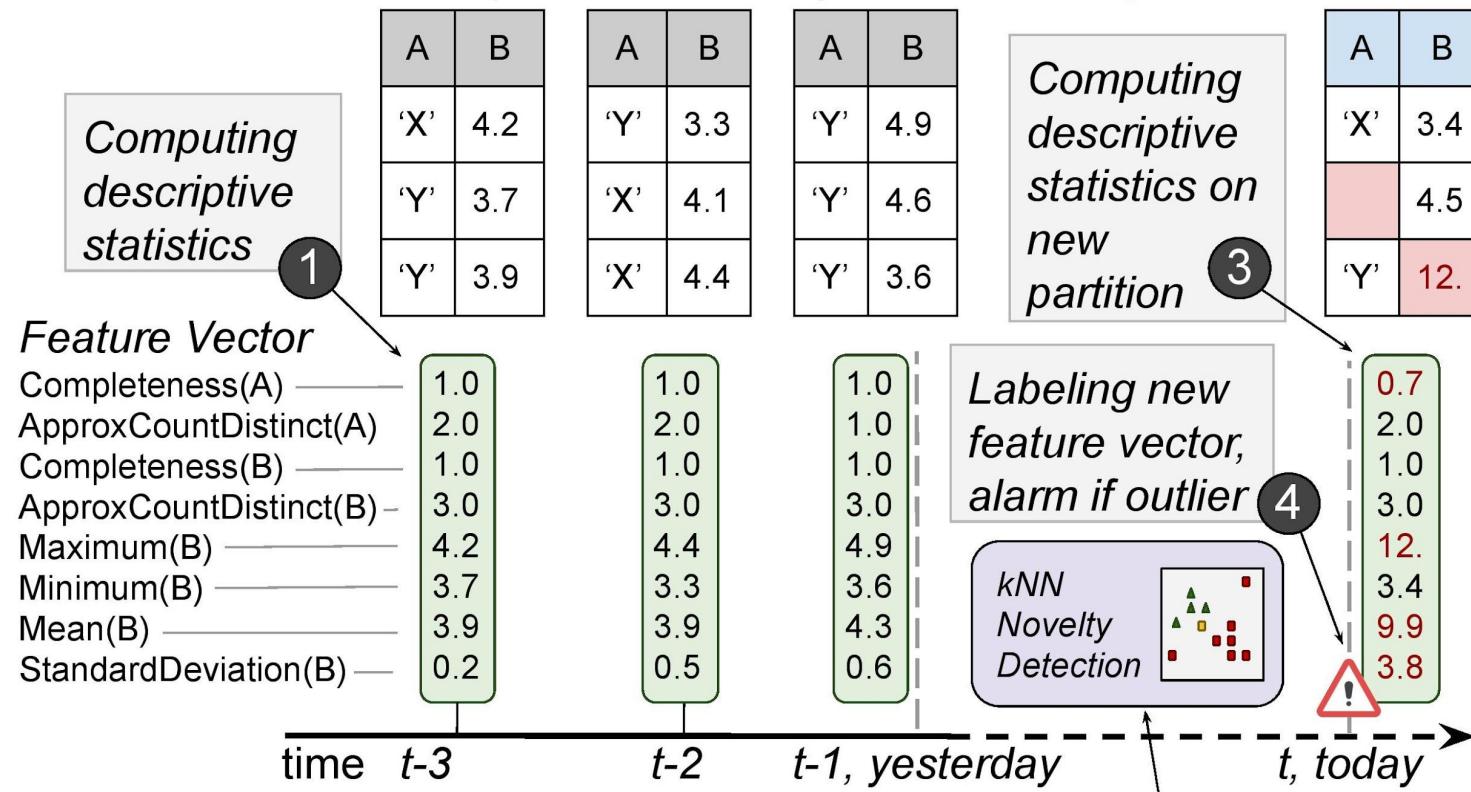
5

```
Targets model
CallFunc DNN -- Args Attr List(Dense(128, input=features.shape),
                                         Dropout(0.25),
                                         Dense(64),
                                         Dense(1, activation='linear'))
```

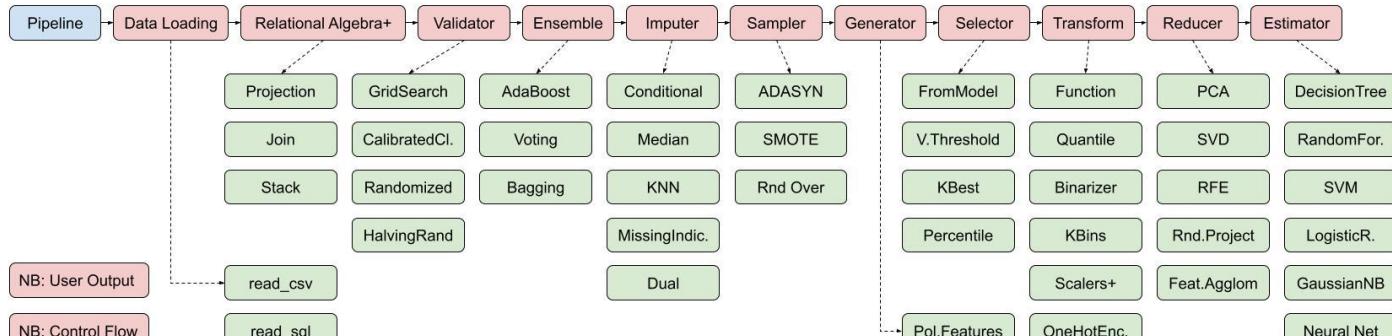
```
Expr CallFunc fit -- Args Attr houses
      Attr houses$price
      Attr model predict(houses)
      Attr Attr houses$price
```

```
Expr CallFunc error -- Args Attr houses
      Attr houses$price
      Attr Attr model predict(houses)
      Attr Attr houses$price
```

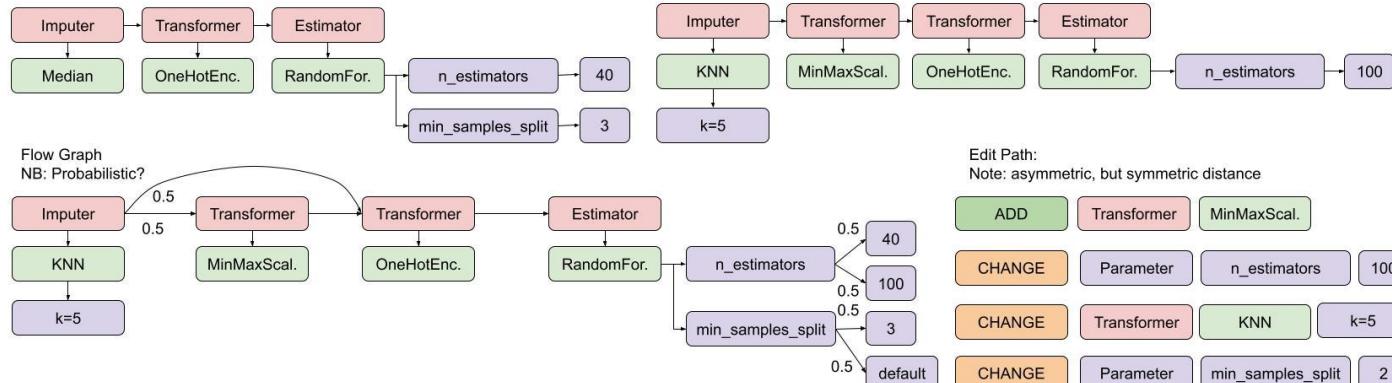
Previously observed data partitions Data partition to validate



Training novelty detection algorithm on feature vectors



Recursive Definition through composition rules, cycles and multiple entry/exit points are implicit. Example below:



Assisted Design of Data Science Pipelines

