



Big Data Analytics Report

Module: IJC427

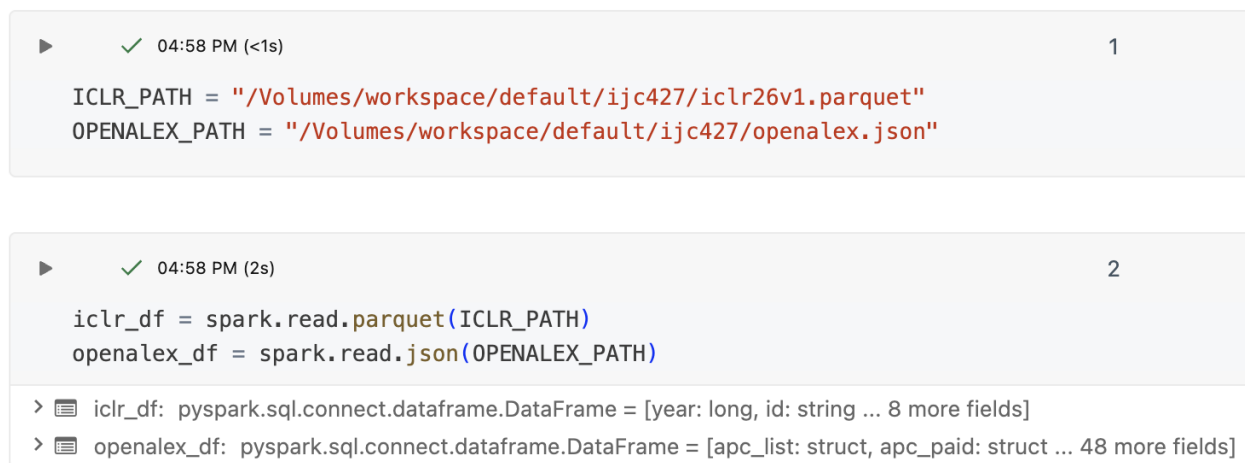
Word Count: 3195

Registration Number: 250214208

1. Description of any setup required

This report presents a big data analytics solution implemented using PySpark DataFrames within the Databricks environment. The analysis was performed using two datasets: an ICLR submission dataset stored in parquet format and an OpenAlex dataset stored in JSON format. The objective of the analysis was to investigate patterns in conference submissions, including acceptance and rejection decisions, reviewer scores, keyword trends and citation impact. Additional analysis was also carried out to explore the relationship between peer-review evaluations and the later academic influence of papers. The analysis was completed from the perspective of a client such as ICLR, who may use these findings to better understand submission trends, evaluate aspects of the review process and identify emerging research topics.

Databricks was used as the primary environment for the implementation of the analytics solution. All analysis was carried out using PySpark DataFrames due to their suitability for distributed processing and large-scale semi-structured data analysis. The ICLR submission dataset was loaded from a parquet file located within a Catalog volume, while the OpenAlex dataset was loaded from a JSON file stored in the same environment. The datasets were accessed using the file paths specified in the assignment brief to ensure reproducibility of the notebook execution. Serverless compute resources provided by Databricks were used to execute the notebook, and all tasks were run within a single Python notebook using the 'Run all' functionality. The implementation relied primarily on Spark DataFrame operations such as filtering, grouping, joins, aggregation and array processing to answer the required analytical questions efficiently.



```
04:58 PM (<1s) 1
ICLR_PATH = "/Volumes/workspace/default/ijc427/iclr26v1.parquet"
OPENALEX_PATH = "/Volumes/workspace/default/ijc427/openalex.json"

04:58 PM (2s) 2
iclr_df = spark.read.parquet(ICLR_PATH)
openalex_df = spark.read.json(OPENALEX_PATH)

> iclr_df: pyspark.sql.connect.dataframe.DataFrame = [year: long, id: string ... 8 more fields]
> openalex_df: pyspark.sql.connect.dataframe.DataFrame = [apc_list: struct, apc_paid: struct ... 48 more fields]
```

Figure 1: Dataset path configuration and loading of the ICLR and OpenAlex datasets

2. Data Cleaning and Preparation

The datasets were inspected using Spark schema operations before analysis was performed. The ICLR dataset contained several fields stored as arrays, including reviewer scores and paper keywords, while the OpenAlex dataset included nested JSON structures such as citation percentile

information. Schema inspection was therefore important to determine the appropriate preprocessing and transformation operations required for each question.

Minimal preprocessing was intentionally used throughout the implementation in order to reproduce the benchmark outputs provided with the assignment brief. This approach was particularly important when joining the ICLR and OpenAlex datasets, where raw title matching was used instead of more advanced normalization or text-cleaning methods. Although more sophisticated preprocessing could potentially improve matching accuracy, the assignment guidance indicated that the provided verification outputs were generated using minimal preprocessing.

```
▶ ✓ 04:58 PM (<1s)

print(iclr_df.count())

iclr_df.printSchema()
openalex_df.printSchema()

> See performance \(1\)
```

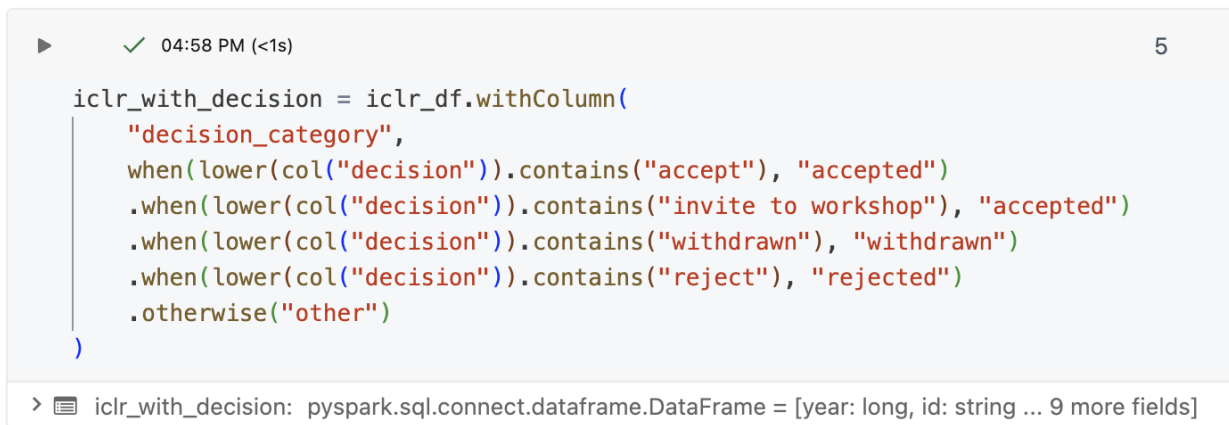
```
55906
root
|-- year: long (nullable = true)
|-- id: string (nullable = true)
|-- title: string (nullable = true)
|-- abstract: string (nullable = true)
|-- authors: string (nullable = true)
|-- author_ids: string (nullable = true)
|-- decision: string (nullable = true)
|-- scores: array (nullable = true)
|   |-- element: long (containsNull = true)
|-- keywords: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- labels: string (nullable = true)

root
|-- apc_list: struct (nullable = true)
|   |-- currency: string (nullable = true)
|   |-- value: long (nullable = true)
|   |-- value_usd: long (nullable = true)
|-- apc_paid: struct (nullable = true)
```

Figure 2: Schema inspection of the ICLR and OpenAlex datasets

Several preprocessing operations were carried out before analysis. Empty reviewer score arrays were filtered when calculating average scores in order to avoid divide-by-zero errors during aggregation. In addition, some papers in the ICLR dataset contained empty decision values, which were excluded from the final decision-count analysis because they could not be reliably classified as accepted, rejected or withdrawn submissions. Keyword arrays were transformed into individual rows using Spark array expansion operations to support yearly frequency analysis and keyword trend comparisons.

PySpark DataFrame operations were used throughout the implementation due to their suitability for scalable distributed data processing. Operations such as filtering, grouping, aggregation, joins and array processing were applied repeatedly across the analytical tasks. The parquet structure of the ICLR dataset also supported efficient column-based access, while the OpenAlex JSON dataset demonstrated the use of Spark for processing semi-structured data.



```
iclr_with_decision = iclr_df.withColumn(
    "decision_category",
    when(lower(col("decision")).contains("accept"), "accepted")
    .when(lower(col("decision")).contains("invite to workshop"), "accepted")
    .when(lower(col("decision")).contains("withdrawn"), "withdrawn")
    .when(lower(col("decision")).contains("reject"), "rejected")
    .otherwise("other")
)

> iclr_with_decision: pyspark.sql.connect.dataframe.DataFrame = [year: long, id: string ... 9 more fields]
```

Figure 3: Decision label preprocessing and categorisation using PySpark

Question 1

Assumptions about the data. It was assumed that decision labels containing the term ‘accept’ represented accepted papers regardless of the specific presentation format, including oral, poster, spotlight and talk presentations. In addition, papers labelled as ‘Invite to Workshop Track’ were treated as accepted submissions because they indicate successful progression into an official conference-related track. Papers labelled as ‘Desk rejected’ were categorized as rejected submissions. Some records contained empty decision values, and these were excluded from the final analysis because they could not be reliably assigned to accepted, rejected or withdrawn categories. It was also assumed that the decision field reflected the final outcome of the review process for each submission.

Implementation. The analysis was implemented using PySpark DataFrame operations within Databricks. A new column named `decision_category` was created using conditional transformations in order to standardize multiple decision labels into broader categories including accepted, rejected and withdrawn. The `lower()` function was used to normalize text values before classification, while `when()` conditions were applied to identify relevant keywords within the decision field. After

preprocessing, the dataset was filtered to exclude uncategorized records and grouped by decision category using Spark aggregation operations to calculate the number of submissions within each category. This approach allowed efficient large-scale processing of the conference submission data.

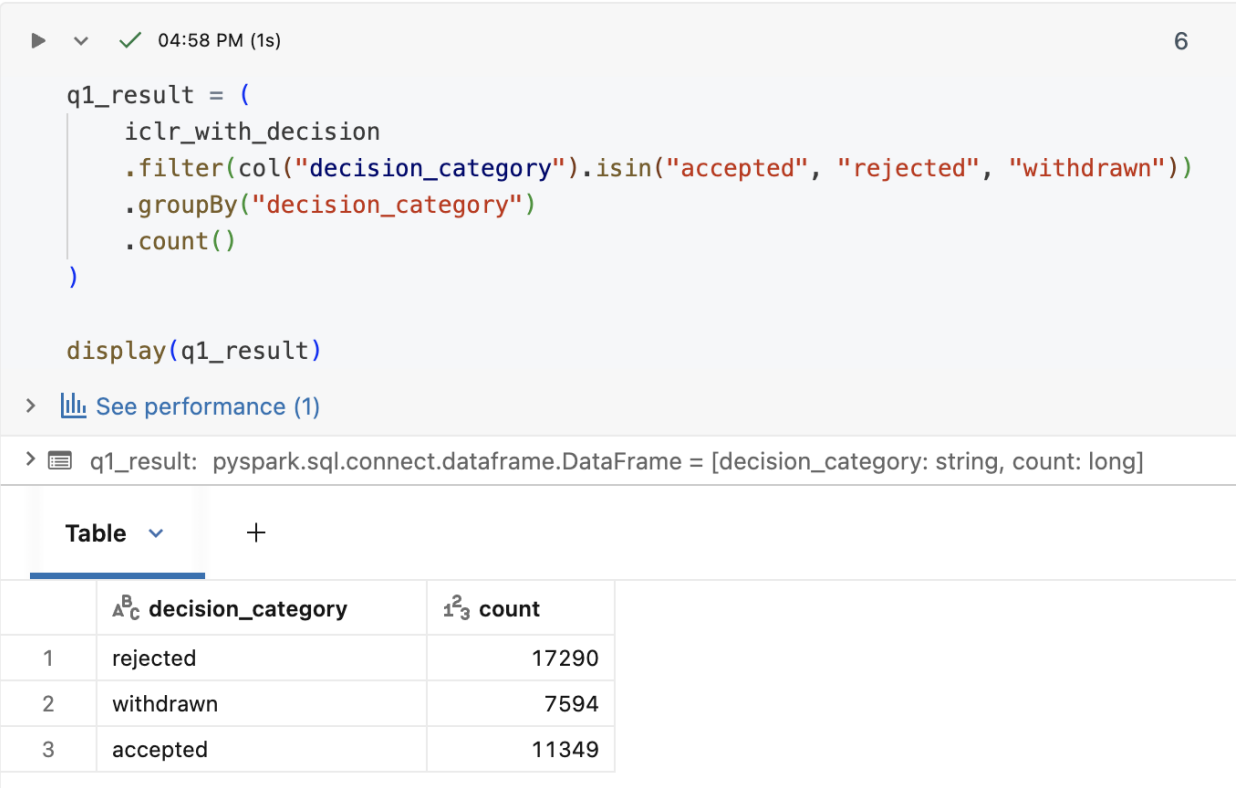


Figure 4: Decision categorisation and aggregation for Question 1

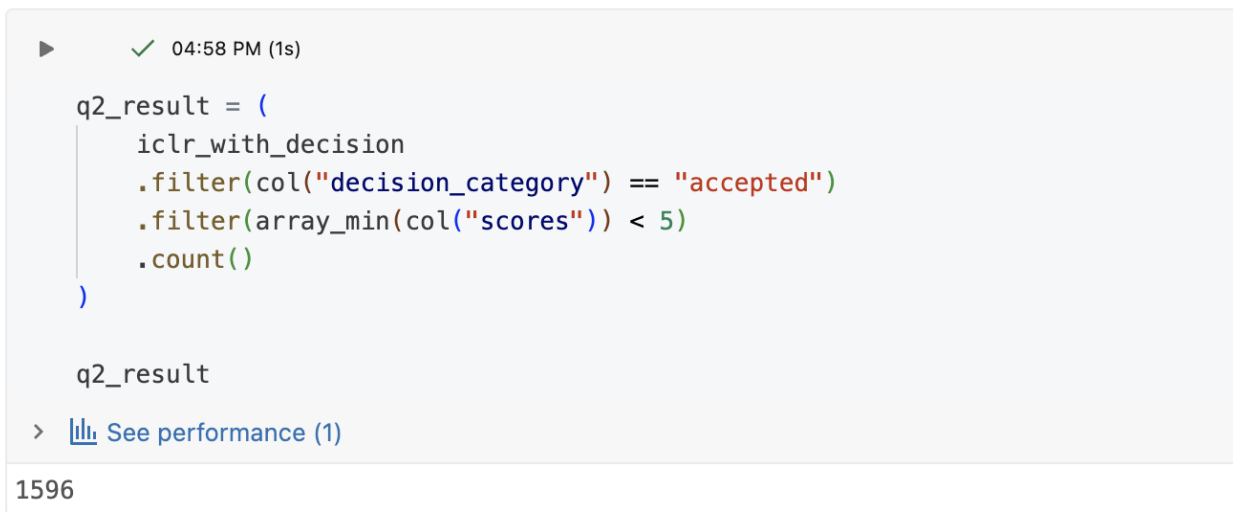
Result on the submission data. The final analysis identified 11,349 accepted papers, 17,290 rejected papers and 7,594 withdrawn papers within the ICLR submission dataset. Records containing empty or unclassified decision values were excluded from the final aggregation.

Discussion. The results show that rejected submissions substantially exceeded accepted submissions, indicating that ICLR is a highly selective conference with strong competition between submitted papers. The large number of rejected papers may reflect both the popularity of the conference and the increasing volume of machine learning research submissions across recent years. In addition, a significant number of withdrawn papers were identified, which may indicate authors withdrawing submissions during the review process, potentially due to concurrent submissions, incomplete work or strategic resubmission decisions. From the client perspective, these findings may help evaluate the scale of reviewer workload and the overall competitiveness of the conference review process. The presence of a large number of submissions also highlights the importance of efficient large-scale review management systems. A possible extension of this analysis would be to investigate yearly acceptance rates or compare acceptance behaviour across different research topics in order to identify whether certain subject areas experience higher acceptance or rejection rates than others.

Question 2

Assumptions. It was assumed that papers categorized as accepted submissions represented successful outcomes of the review process. Reviewer score arrays were assumed to contain valid numerical reviewer evaluations for each paper. A score below 5 was treated as a relatively weak reviewer assessment based on the scoring scale used within the dataset. Papers containing empty reviewer score arrays were excluded from the analysis because it was not possible to determine whether they satisfied the scoring condition.

Implementation. The analysis was implemented using PySpark DataFrame filtering and aggregation operations. The previously generated decision categories were reused to identify accepted papers only. The `array_min()` function was then applied to the reviewer score arrays in order to identify papers containing at least one score below 5. After filtering the dataset according to these conditions, Spark aggregation operations were used to calculate the total number of matching papers. Using array-based functions within PySpark allowed the reviewer score data to be processed efficiently without converting the arrays into separate rows.



```
▶ 04:58 PM (1s)

q2_result = (
    iclr_with_decision
    .filter(col("decision_category") == "accepted")
    .filter(array_min(col("scores")) < 5)
    .count()
)

q2_result

> See performance \(1\)

1596
```

Figure 5: Identification of accepted papers containing reviewer scores below 5

Result. The analysis identified 1,596 accepted papers that contained at least one reviewer score below 5.

Discussion. The results demonstrate that a substantial number of accepted papers received at least one relatively weak reviewer evaluation despite ultimately being accepted. This suggests that the conference review process does not rely solely on unanimous reviewer agreement and may instead consider broader factors such as reviewer discussion, paper novelty, significance or disagreement between reviewers. The findings may also indicate that some papers generated mixed opinions among reviewers while still being considered sufficiently valuable for acceptance. From the client perspective, this analysis may help identify the extent of reviewer disagreement within the peer-review process and highlight the complexity of evaluating research quality. A possible extension

of this analysis would be to examine the distribution of reviewer scores for accepted and rejected papers in more detail, or to investigate whether papers with highly variable reviewer scores later achieve strong citation impact.

Question 3

Assumptions. It was assumed that reviewer scores were numerical and directly comparable across different papers and years within the dataset. The mean reviewer score was treated as a reasonable representation of the overall reviewer evaluation for each paper. Papers containing empty reviewer score arrays were excluded because average scores could not be calculated reliably. It was also assumed that higher mean scores reflected stronger reviewer support and more positive overall evaluations.

Implementation. The analysis was implemented using PySpark DataFrame transformations and aggregation functions. Papers containing non-empty reviewer score arrays were first identified to ensure valid average score calculations. Spark aggregation operations were then applied to calculate the mean reviewer score for each paper using array-based summation and division by the number of reviewer scores. The resulting papers were ordered in descending order according to their mean scores, and the top ten highest-scoring papers were selected along with their paper identifiers and publication years. This approach allowed efficient processing of reviewer score arrays without requiring manual iteration over individual values.

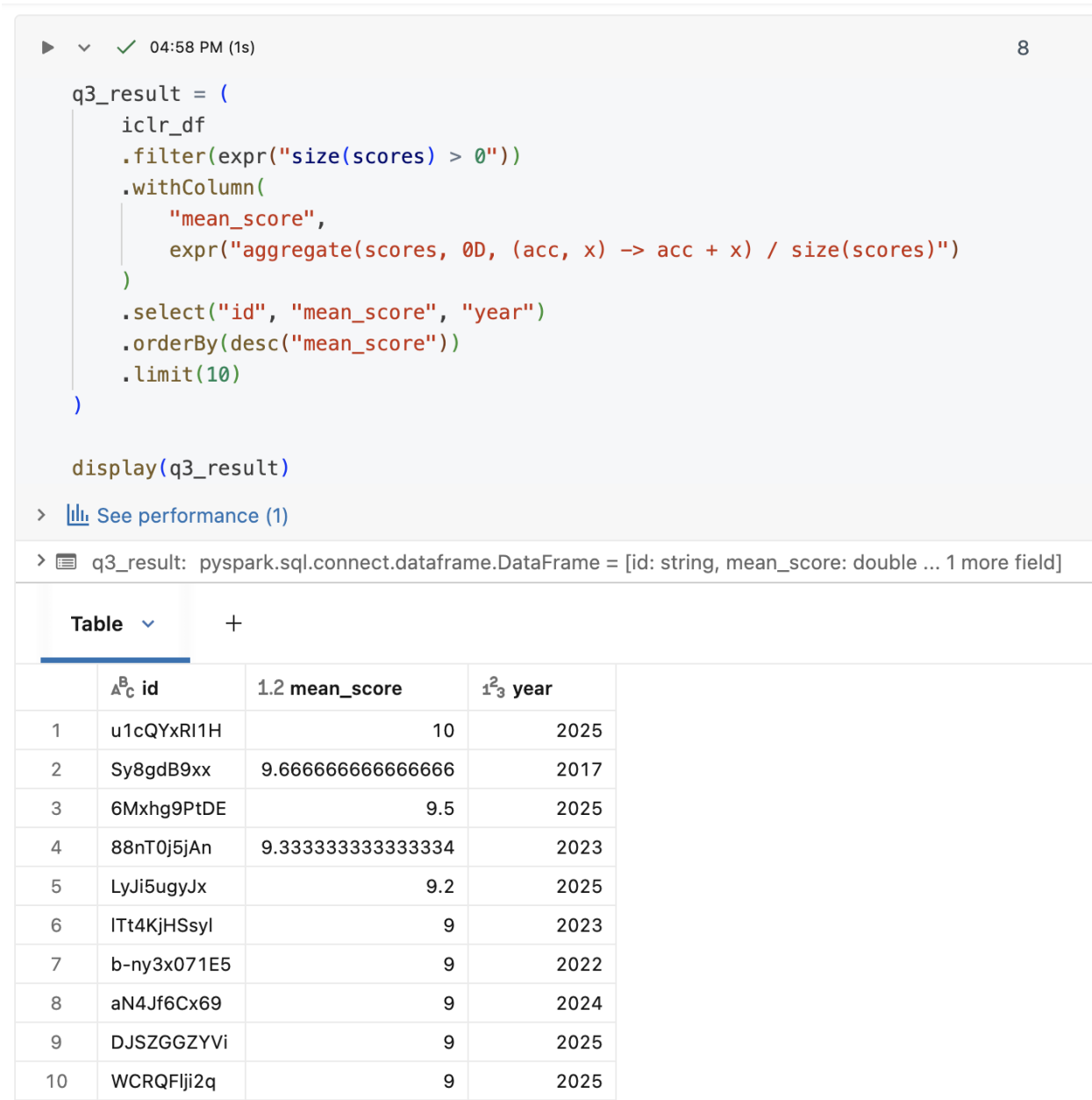


Figure 6: Highest mean reviewer scores within the ICLR dataset

Result. The analysis identified the ten papers with the highest mean reviewer scores across the ICLR dataset. Several of the highest-scoring papers were associated with recent submission years, including multiple papers from 2025, while the highest recorded mean score was equal to 10.

Discussion. The results indicate that several papers achieved extremely strong reviewer evaluations, including perfect or near-perfect mean scores. The presence of multiple highly rated papers in recent years may reflect increasing specialization within certain research areas or stronger alignment between reviewer expectations and submission quality. High mean reviewer scores may also indicate strong consensus among reviewers regarding the novelty, technical

quality or significance of particular papers. From the client perspective, identifying highly rated papers may help highlight submissions that are likely to attract significant research attention or represent important contributions within the field. However, reviewer scores alone may not fully predict long-term research impact, as later citation analysis demonstrated that some lower-scoring papers also achieved strong academic influence. A possible extension of this analysis would be to compare reviewer scores with future citation counts in order to investigate the relationship between peer-review evaluation and long-term research impact in more detail.

Question 4

Assumptions. It was assumed that the keywords submitted by authors accurately represented the primary research topics of their papers. Keywords appearing in different years were treated as directly comparable, although minor inconsistencies in spelling or formatting may exist within the dataset. The analysis assumed that a large increase in keyword frequency between consecutive years indicated rapid growth in the popularity or relevance of a particular research topic. Each keyword within the keyword arrays was treated as an independent observation after preprocessing.

Implementation. The analysis was implemented using PySpark DataFrame transformations and aggregation operations. Keyword arrays within the ICLR dataset were expanded into individual rows using Spark array expansion functions in order to allow keyword-level frequency analysis. The transformed dataset was then grouped by year and keyword to calculate yearly keyword frequencies. A self-join operation was applied to compare keyword frequencies between consecutive years, allowing identification of keywords that appeared fewer than five times in one year but more than fifty times in the following year. The resulting keywords were ordered according to their frequency in the following year to identify the strongest increases in topic popularity.

▶ ✓ 04:58 PM (<1s)

```
keyword_counts = (  
    iclr_df  
    .select("year", explode(col("keywords")).alias("keyword"))  
    .groupBy("year", "keyword")  
    .count()  
)
```

>  keyword_counts: pyspark.sql.connect.dataframe.DataFrame = [year: long, keyword: string ... 1 more field]

▶ ✓ 04:58 PM (1s)

```
kc_current = keyword_counts.alias("current")  
kc_next = keyword_counts.alias("next")  
  
q4_result = (  
    kc_current  
    .join(  
        kc_next,  
        (col("current.keyword") == col("next.keyword")) &  
        (col("next.year") == col("current.year") + 1),  
        "inner"  
    )  
    .filter(col("current.count") < 5)  
    .filter(col("next.count") > 50)  
    .select(  
        col("current.keyword").alias("keyword"),  
        col("current.year").alias("year"),  
        col("current.count").alias("count_first_year"),  
        col("next.year").alias("following_year"),  
        col("next.count").alias("count_following_year")  
    )  
    .orderBy(desc("count_following_year"))  
    .limit(10)  
)  
  
display(q4_result)
```

>  See performance (1)

>  kc_current: pyspark.sql.connect.dataframe.DataFrame = [year: long, keyword: string ... 1 more field]

> q4_result: pyspark.sql.connect.dataframe.DataFrame = [keyword: string, year: long ... 3 more fields]

	keyword	year	count_first_year	following_year	count_following_year
1	test-time scaling	2025	3	2026	87
2	llm	2023	4	2024	76
3	large reasoning mod...	2025	1	2026	57
4	spatial reasoning	2025	3	2026	57

Figure 7: Identification of rapidly increasing keywords across consecutive years

Result. The analysis identified ‘llm’ as the most significant rapidly emerging keyword within the dataset. The keyword appeared only four times during 2023 but increased to seventy-six occurrences in 2024, satisfying the conditions specified in the analytical task.

Discussion. The results demonstrate a rapid increase in the use of the keyword ‘llm’, indicating a major shift in research focus toward large language model technologies within the machine learning community. The substantial growth between 2023 and 2024 likely reflects the widespread influence of modern generative AI systems and the increasing research interest surrounding transformer-based language models during this period. From the client perspective, identifying rapidly emerging topics may help conference organizers understand changing research priorities and anticipate future submission trends. Such insights may also support reviewer allocation, workshop planning and topic-specific conference organization. The results additionally demonstrate how keyword-based analysis can be used to monitor the evolution of research areas over time. A possible extension of this analysis would be to investigate longer-term keyword growth trends or compare emerging topics across different subfields of machine learning.

Question 5

Assumptions. It was assumed that paper titles were sufficiently consistent between the ICLR and OpenAlex datasets to allow direct matching using raw title values. Papers sharing the same title across both datasets were treated as representing the same research work. The analysis also assumed that unmatched papers were either absent from the OpenAlex subset or contained title formatting differences preventing successful joins. Minimal preprocessing was intentionally maintained in order to reproduce the benchmark outputs specified in the assignment guidance.

Implementation. The analysis was implemented using PySpark DataFrame join and aggregation operations. An inner join was applied between the ICLR dataset and the OpenAlex dataset using paper titles as the matching field. Only papers successfully matched across both datasets were retained within the joined dataset. The resulting records were then grouped according to the ICLR submission year, and Spark aggregation operations were used to calculate the number of matched papers within each year. The implementation demonstrated the use of distributed join operations for integrating information across multiple large-scale datasets.

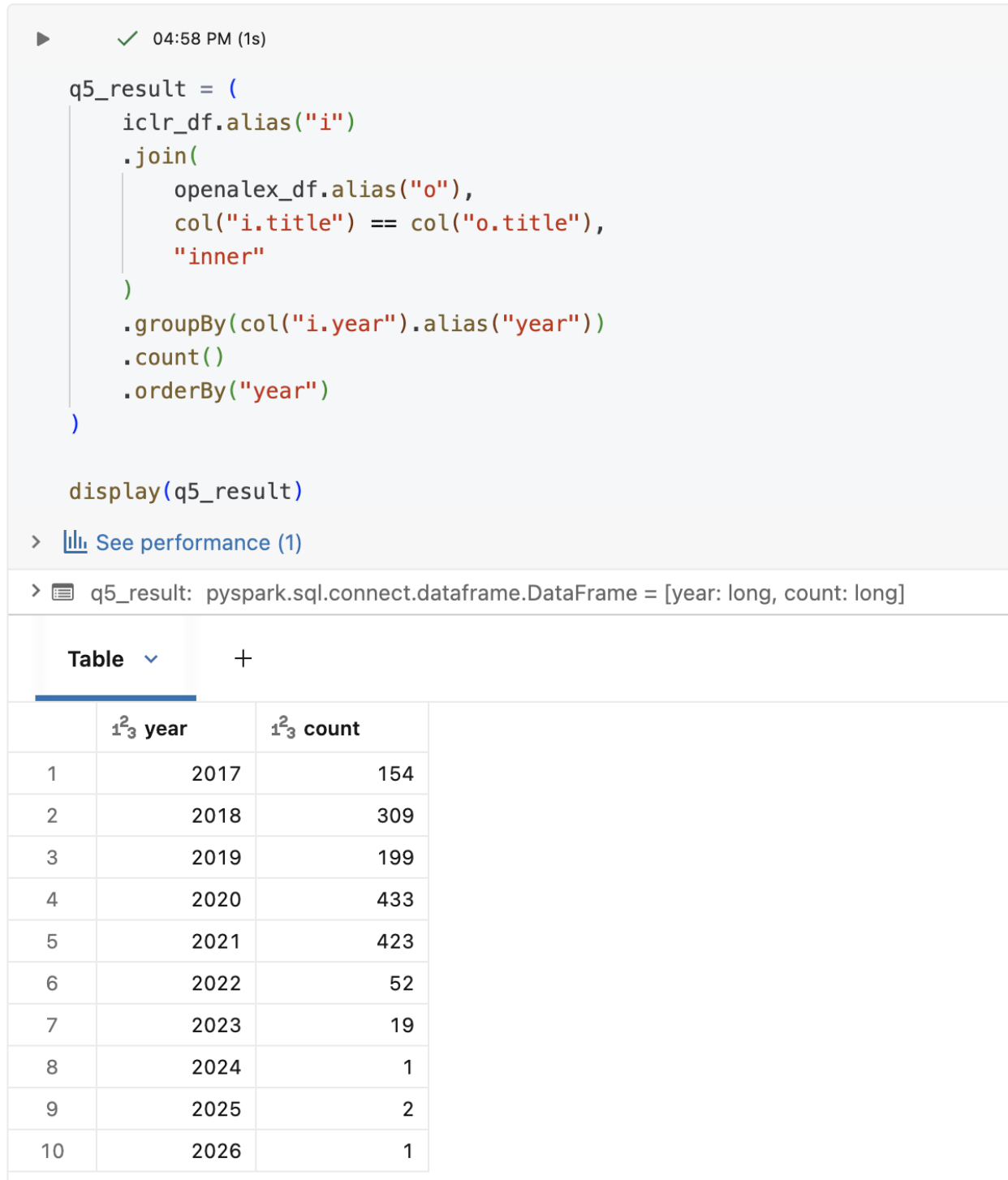


Figure 8: Matching ICLR submissions with OpenAlex records by year

Result. The analysis identified successful matches between the ICLR and OpenAlex datasets across multiple years. The highest number of matched papers occurred in 2020 and 2021, while smaller numbers of matches were observed for more recent years including 2024, 2025 and 2026.

Discussion. The results demonstrate that a substantial number of ICLR submissions could be linked to corresponding OpenAlex records, allowing additional citation and publication metadata to be incorporated into the analysis. The higher number of matches observed for earlier years is likely related to the greater availability of indexed publication information and citation data for older papers. In contrast, more recent papers may not yet be fully indexed within OpenAlex or may not yet have accumulated sufficient publication metadata. From the client perspective, integrating external bibliographic datasets may provide additional insight into the long-term impact and visibility of conference submissions. The results also demonstrate the practical challenges associated with dataset integration, particularly when relying on raw title matching without extensive normalization. A possible extension of this analysis would be to implement more advanced matching approaches, such as DOI-based matching or text similarity techniques, in order to improve match coverage and reduce the impact of title formatting inconsistencies.

Question 6

Assumptions. It was assumed that the citation percentile information provided by OpenAlex represented a reliable indicator of long-term academic impact. Papers with citation percentile values greater than or equal to 0.95 were treated as highly influential publications within the broader research landscape. Reviewer scores were assumed to be directly comparable across papers, while papers with mean reviewer scores below 5 were interpreted as receiving relatively weak reviewer evaluations. It was also assumed that title-based matching between the ICLR and OpenAlex datasets correctly linked the corresponding research papers.

Implementation. The analysis was implemented using PySpark DataFrame joins, filtering operations and array-based aggregation functions. The ICLR dataset was joined with the OpenAlex dataset using paper titles in order to combine reviewer evaluation data with citation-based metrics. Papers containing empty reviewer score arrays were excluded before calculating mean reviewer scores using Spark aggregation operations applied to the score arrays. The OpenAlex field `citation_normalized_percentile.value` was then used to identify papers within the top five percent of citation impact. Finally, papers with mean reviewer scores below 5 and citation percentile values greater than or equal to 0.95 were selected as the final result set. This implementation combined peer-review data and bibliometric indicators within a single distributed analytical workflow.

```
q6_result = (  
  iclr_df.alias("i")  
    .join(  
      openalex_df.select(  
        "title",  
        "cited_by_count",  
        "citation_normalized_percentile"  
      ).alias("o"),  
      col("i.title") == col("o.title"),  
      "inner"  
    )  
    .filter(expr("size(i.scores) > 0"))  
    .withColumn(  
      "mean_score",  
      expr("aggregate(i.scores, 0D, (acc, x) -> acc + x) / size(i.scores)")  
    )  
    .filter(col("mean_score") < 5)  
    .filter(col("citation_normalized_percentile.value") >= 0.95)  
    .select(  
      col("i.id").alias("id"),  
      col("i.year").alias("year"),  
      col("i.title").alias("title"),  
      col("mean_score"),  
      col("cited_by_count"),  
      col("citation_normalized_percentile.value").alias("citation_percentile")  
    )  
    .distinct()  
    .orderBy("year", "id")  
)  
  
display(q6_result)  
  
q6_result.count()
```

> q6_result: pyspark.sql.connect.dataframe.DataFrame = [id: string, year: long ... 4 more fields]

	id	year	title	mean_score	cited_by_count	citation_percentile
1	S19eAF9ee	2017	Structured Sequence Modeling with Graph Convolutional Recurrent Networks	4	763	0.99934892
2	BkPrDFgR-	2018	Piecewise Linear Neural Networks verification: A comparative study	4.666666666666667	49	0.9762242
3	H1bM1fZCW	2018	GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks	4.666666666666667	313	0.99770379
4	HJWGdbbCW	2018	Reinforcement and Imitation Learning for Diverse Visuomotor Skills	4.666666666666667	209	0.98738113
5	HkmaTz-0W	2018	Visualizing the Loss Landscape of Neural Nets	4.666666666666667	514	0.99945664
6	Hy-IxYDWG	2018	Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing	3.3333333333333335	128	0.98713535
7	HJeq43AqF7	2019	Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Auto-Encoders	4.333333333333333	111	0.9879825
8	B1esx6EYvr	2020	A critical analysis of self-supervision, or what we can learn from a single image	4.333333333333333	51	0.96159674
9	ByexEISYDr	2020	Fair Resource Allocation in Federated Learning	4	132	0.99137782
10	Byg5fIHfDr	2020	EvoNet: A Neural Network for Predicting the Evolution of Dynamic Graphs	2.3333333333333335	13	0.97119179
11	H1emfT4twB	2020	Few-shot Text Classification with Distributional Signatures	3.3333333333333335	54	0.9627522
12	HylvlaVtwr	2020	Learning to Sit: Synthesizing Human-Chair Interactions via Hierarchical Control	2.3333333333333335	20	0.98873944
13	S1xFm6VKDH	2020	Meta Module Network for Compositional Visual Reasoning	3	60	0.95795698

Figure 9: Identification of low-scoring papers with high citation impact

Result. The final analysis identified 13 papers that had a mean reviewer score below 5 but were within the top 5% of citation impact according to the OpenAlex citation percentile field. The identified papers were mainly from earlier submission years, particularly between 2017 and 2020.

Discussion. The results show that some papers with relatively weak reviewer evaluations later achieved strong academic impact. This suggests that reviewer scores do not always fully predict the long-term influence of a research paper. Some papers may initially appear risky, niche or technically uncertain, but later become important because they introduce useful methods, address emerging problems or influence future research directions.

From the client perspective, this finding is important because it highlights a potential gap between short-term peer-review judgement and long-term research value. It does not necessarily mean that the review process is ineffective, but it shows that conference organizers may benefit from monitoring how accepted, rejected or low-scoring papers perform after publication. A useful extension would be to compare citation impact across accepted and rejected papers, or to examine whether papers with high reviewer disagreement are more likely to become highly cited later.

Extension: Reviewer Disagreement and Long-Term Citation Impact

Motivation for the extension. The earlier analysis showed that some papers with relatively low reviewer scores later achieved strong citation impact. This raised a further question about whether papers that divide reviewer opinion may sometimes become influential after publication. For a client such as ICLR, this is valuable because highly novel or unconventional papers may not always receive consistent reviewer evaluations during the review process. Investigating reviewer disagreement can therefore provide additional insight into whether controversial papers later become important within the research community.

Implementation outline. This extension measured reviewer disagreement by calculating the spread between the highest and lowest reviewer score for each paper. Papers with fewer than two reviewer scores were excluded because disagreement could not be meaningfully calculated. The

resulting score spread was then joined with OpenAlex citation data using paper titles. Papers were identified as high-impact if their OpenAlex citation percentile was at least 0.95. The final extension result selected papers with a large reviewer score spread and high citation impact, allowing the analysis to focus on submissions that were both controversial during review and influential after publication.

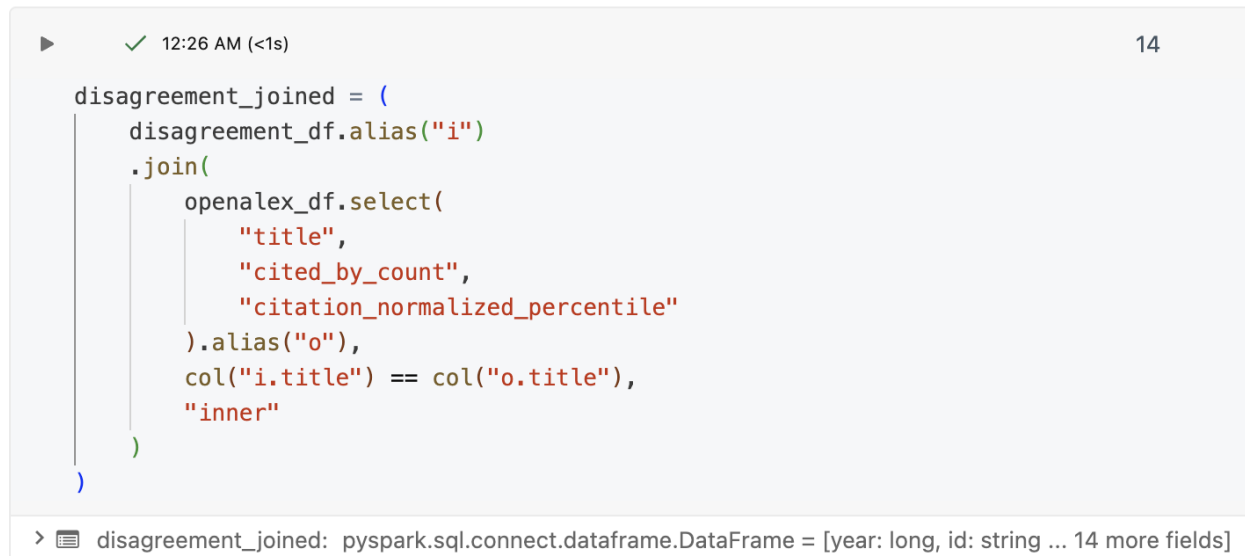


```
disagreement_df = (
    iclr_df
    .filter(expr("size(scores) > 1"))
    .withColumn(
        "max_score",
        array_max(col("scores"))
    )
    .withColumn(
        "min_score",
        array_min(col("scores"))
    )
    .withColumn(
        "score_spread",
        col("max_score") - col("min_score")
    )
)
```

>  disagreement_df: pyspark.sql.connect.dataframe.DataFrame = [year: long, id: string ... 11 more fields]

Figure 10: Calculation of reviewer score spread using PySpark array operations

Result of the extension. The extension identified papers that had large differences between reviewer scores but later achieved high citation impact. This result suggests that some papers receiving mixed reviewer evaluations may still become influential within the research community. In practical terms, these papers may have been difficult to assess during peer review because they involved new ideas, unfamiliar methods or research directions that were not yet widely accepted.



```
12:26 AM (<1s) 14

disagreement_joined = (
    disagreement_df.alias("i")
    .join(
        openalex_df.select(
            "title",
            "cited_by_count",
            "citation_normalized_percentile"
        ).alias("o"),
        col("i.title") == col("o.title"),
        "inner"
    )
)

> [ ] disagreement_joined: pyspark.sql.connect.dataframe.DataFrame = [year: long, id: string ... 14 more fields]
```

Figure 11: Integration of reviewer disagreement data with OpenAlex citation metadata

Discussion of extension. This extension provides a stronger insight than simply counting accepted and rejected papers because it links peer-review behaviour with long-term research influence. A high score spread suggests that reviewers did not fully agree about the quality or value of a paper. If such papers later become highly cited, this may indicate that disagreement is not always a weakness in the review process; instead, it may sometimes signal innovative or uncertain work with future impact.

From the client perspective, this finding could support more careful consideration of papers with strong reviewer disagreement. Rather than treating disagreement only as a negative signal, conference organizers could examine whether these papers require additional discussion, specialist reviewers or clearer decision-making processes. This may help reduce the risk of undervaluing papers that are initially controversial but later become influential. A further extension would be to compare disagreement patterns between accepted and rejected papers, or to investigate whether particular research topics are more likely to produce high reviewer disagreement and high future citation impact.

```
✓ 12:26 AM (1s) 15

extension_result = (
    disagreement_joined
    .filter(col("score_spread") >= 4)
    .filter(col("citation_normalized_percentile.value") >= 0.95)
    .select(
        col("i_id").alias("paper_id"),
        col("i_year"),
        col("i_title"),
        col("score_spread"),
        col("o_cited_by_count")
    )
    .orderBy(desc("cited_by_count"))
)

display(extension_result)

> See performance \(1\) Optimize

> extension_result: pyspark.sql.connect.dataframe.DataFrame = [paper_id: string, year: long ... 3 more fields]
```

	^A _C paper_id	¹ ₃ year	^A _C title	¹ ₃ score_spread	¹ ₃ cited_by_count
19	H1emfT4twB	2020	Few-shot Text Classification with Distributional Signatures	5	54
20	BJ8c3f-0b	2018	Auto-Encoding Sequential Monte Carlo	4	54
21	ByRWCqvT-	2018	Learning to cluster in order to transfer across domains and tasks	4	52
22	B1esx6EYvr	2020	A critical analysis of self-supervision, or what we can learn from a single image	5	51
23	qrwe7XHTmYb	2021	GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding	5	49
24	HJgcvJBfVb	2020	Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning	5	47
25	rkaT3zWCZ	2018	Building Generalizable Agents with a Realistic and Rich 3D Environment	4	40
26	HkxBJT4YvB	2020	Learning Disentangled Representations for Counterfactual Regression	5	40
27	B1hYRMbCW	2018	On the regularization of Wasserstein GANs	5	37
28	3k20LAiHYL2	2021	Pre-training Text-to-Text Transformers for Concept-centric Common Sense	4	35
29	BJg4Z3RqF7	2019	Unsupervised Adversarial Image Reconstruction	4	26
30	rkgHYONYwr	2020	Discovering Motor Programs by Recomposing Demonstrations	5	22
31	H1Dkn7ZCZ	2018	Deep learning mutation prediction enables early stage lung cancer detection in liquid biopsy	4	20
32	H1lmyRNFvr	2020	Augmenting Genetic Algorithms with Deep Neural Networks for Exploring the Chemical Space	5	20
33	rvl5khRcKm	2019	Human-level Protein Localization with Convolutional Neural Networks	4	10

Figure 12: Highly cited papers with strong reviewer disagreement