Unveiling Experts in Data Science: A Mining Software Repository Perspective

José Ferreira[1], Johnatan Oliveira[2], Eduardo Figueiredo[1]

[1]Department of Computer Science, Federal University of Minas Gerais (UFMG)
Belo Horizonte-MG, Brazil
[2]Department of Computer Science, Federal University of Lavras (UFLA/ICTIN)
São Sebastião do Paraíso-MG, Brazil
josenicuri@ufmg.br, johnatan.oliveira@ufla.br, figueiredo@dcc.ufmg.br

Abstract

Data science is a field of knowledge that exploits various methods of collecting and analyzing data. It is increasingly replacing traditional computational methods to explore new software frontiers, particularly when data become the most valuable resource. Nowadays, data-driven software application development is getting more common and multiple experts take part of the process, such as data scientists, alongside software developers. Skills of data science professionals are basically pillar for projects. Nevertheless, locating someone who possesses strong technical skills as a data scientist may be a difficult task because such data are not easy to verify. This paper addresses this problem by exploring the activity of software repositories, such as commits, to identify technical skills of data scientists. The provided data consisted of the results accumulated from 18 data science projects stored on GitHub that satisfied certain criteria, such as number of stars, creation date, and commit frequency. By analyzing these projects, we identified a total of 69 data scientists. Accordingly, a data science proficiency measurement was then conducted by counting and categorizing the number and types of changes applied based on the metrics and profiles created. In such situation, the outcome shows that Python is the most opted programming language by data scientists.

keywords: Data Science, Mining Repositories, Expert Identification.

1 Introduction

Data are the new currency in our modern era. This trend is transforming software development into a data-oriented sphere. Hence, the association of data scientists with software engineers by their nature has to be indispensable for the projects success [1]. Data science professionals know about the process of data science, and this is the secret to the most successful software development projects. Nevertheless, the recognition of their skills is rather a complex issue, being not a simple task [5].

The review of relevant literature shows a number of crucial studies [1, 2],concerning software engineering and data science. For instance, Kim et al. [5] explored the dynamic role of data scientists at Microsoft and pinpointed the growing influence of data science in the context of software development processes. With a different focus, Saltz et al. [8] examined the shift of software engineers into data engineering positions. In a different direction, Oliveira et al. [7] developed source code analysis models that are able to identify programming skills, but they do not focus on data science expertise. Our work tries to extend the previous studies by mining data science skills and measuring the level of proficiency in data science among the contributors to software repositories. In this regard, our goal is to provide valuable perspectives on the dynamic and ever-changing nature of software engineering and the growing demand of data science skills, which now need to be integrated.

This paper conducts a systematic investigation of data science competencies as an aspect of software development, through the application of a structured method to find and assess the skills of contributors in 18 data science projects. This research is conducted in three main phases. In the first phase, the features of both projects and developers were strictly chosen taking into account their relevance and representation, thus a pertinent and representative sample was collected. In the second phase, data extraction approaches were performed to create the study dataset. For this dataset, data science projects provided detailed information on the frequency and nature of the modifications made on a regular basis. The final phase included data analysis of the gathered data, leading to the stratification of developers who demonstrated the best involvement in data science projects. As a result of the exploratory analysis, the study was able to discover important patterns and to provide more detailed information concerning the level of expertise of these contributors, revealing the gradual increase in the role of data science skills in software development.

We observed that selected data scientists [1]had a very high level of commitment to their projects, each making an average of 775 commits in two years. Besides, our research uncovered Python's dominance in data science projects, which

[1]In this paper, we define "data scientists" as those who develop code in data science projetcs.

further exemplifies the greater role that Python plays in the field when looking at the analyzed projects. Python had a total of 61.209 files modified, while TypeScript which comes in second place had almost half that amount, 30.691.

The paper is organized as follows. Section 2 outlines the research purpose. Section 3 presents the results. Section 4 addresses potential validity threats. Section 5 reviews related work. Finally, Section 6 summarizes key findings and suggests future research directions.

2 Study Design

2.1 Goal and Research Questions

This paper focuses on the discovery of fundamental skills of data science professionals. Its primary goal is to provide an outline and description of experts in data-based fields who have had a major positive impact on data science projects. To achieve this, we identify relevant repositories using metrics, such as the number of lines edited, the number of commits made, and the number of files edited by contributors. The following research questions were formulated for the purpose of aligning the goal with our results (RQs).

RQ1: *How the selected metrics provide information on the averages of data scientists' individual contributions*
RQ2: *What are the characteristics of individuals identified as experts in data science projects?*
RQ3: *Which programming languages are mostly used by data science experts?*

2.2 Evaluation Steps

We select 18 GitHub data science projects (see Table 1). This dataset is used as a guide when choosing relevant data science specialists for additional examination. We want to make sure the dataset is inclusive and representative, including a broad range of data science activities across many domains and development processes. Therefore, we are looking at a variety of repositories. Next, we examine the dataset that we used in our study, utilizing metrics that were modified for evaluating data science expertise and were inspired by prior research [7, 13]. Developer activity levels, code contributions, and the extent of their involvement in the chosen projects are all shown by these indicators. We employ statistical approaches and visualization techniques to analyze the retrieved data. Through an evaluation process, we aim to uncover meaningful patterns, identify skilled data scientists, and gain deeper insights into the landscape of data science expertise.

Table 1: Used GitHub repositories

Repository	Stars	Contributors
OpenMined/PySyft	9.2k	423
kedro-org/	9.3k	211
goplus/gop	8.8k	39
Netflix/metaflow	7.5k	88
google/deepvariant	3.1k	24
quadratichq/quadratic	2.7k	22
colour-science/colour	1.9k	45
NannyML/nannyml	1.7k	29
apache/systemds	1k	180
visualpython/visualpython	799	6
LineaLabs/lineapy	653	21
googleapis/python-aiplatform	520	93
IBM/lale	320	25
nebari-dev/nebari	254	63
EpistasisLab/Aliro	219	20
mithril-security/bastionlab	165	12
vertica/VerticaPy	214	16
microsoft/MLOS	123	18

2.3 Dataset

From a starting pool of 629 projects that were automatically searched on GitHub, we chose a subset of 18. By utilizing the "explorer" tab and the keyword "Data Science stars:>100", we were able to locate projects that had more than 100 stars and a specific level of significance. A wide variety of projects in terms of size, complexity, and application domains were guaranteed by our criteria. Our goal is to identify the top 20% of contributors who are highly engaged with theses projects. By focusing on these key contributors, we improve the research process for categorizing data science specialists and optimize resource allocation. Figure 1 shows the selection criteria, as we explain in the following.

Step 1: We looked for repositories with more than 100 stars on GitHub, indicating community engagement and project impact, after searching for "data science".

Step 2: We set a time restriction, requiring repository to exist for a minimum of two years. This guarantees the stability and maturity of the project, enabling a thorough examination of each project's development over time.

Step 3: We pre-processed to identify projects with high development activity, choosing those with more than 10 active contributors to guarantee a representative sample of contributors' involvement.

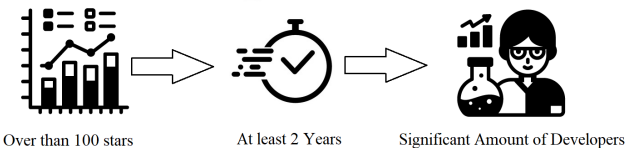

Figure 1: Steps for Criteria Evaluation

2.4 Used Technology and Metrics

We used the Python framework "Pydriller" [11] to extract and analyze data from software repositories.

We conduct a thorough study of the version control system for the project and extract key metrics pertaining to the contributions of the data scientists. We were able to collect data from Pydriller, including developer emails, the quantity of commits, commit messages, and the number of lines added and removed for each commit. Pydriller provided specifics on the files that each developer had edited, giving an understanding of the parts of the codebase that each developer had worked on. The used metrics are: number of commits, lines of code (LOC), and changed files. These metrics are based on related literature [7]. As explained below, these metrics were selected because they can measure various aspects of a developer's involvement and expertise in the project.

Number of Commits. This metric provides insights into the frequency and extent of contributions made by contributors throughout the project's lifecycle. We collect this metric by iterating through the repository's commit history and recording the number of commits made by each contributor. The resulting dataset provides a quantitative measure of a contributor's active participation in the project over time.

Lines of Code. Quantifying the volume of code contributions made by contributors is essential for understanding their impact on the project's codebase. To achieve this, we analyzed the lines of code added and removed by each contributors across the project's commits. By iterating through the commit history and examining the changes introduced in each commit, we calculated the total number of lines added and removed by each contributor.

Number of Changed Files. To capture this aspect, we analyzed the number of files modified by each contributor. By parsing through the commit history and scrutinizing the changes introduced in each commit, we count of files modified by each contributor. The resulting dataset offers insights into the diversity of a contributors's contributions, highlighting their involvement across different components or aspects of the project. Additionally, by leveraging file extensions, we were able to discern the programming languages associated with the modified files, providing supplementary context to the developers' contributions. For example, if data scientists change .py files often, we assign the Python language to their expertise.

By considering these 3 metrics together, we aim to comprehensively evaluate the expertise and contributions of data scientists within data science projects. Together, they provide a multifaceted perspective on contributors' activity levels, coding skills, and impact on project development. This holistic approach allows us to identify and recognize data science professionals who demonstrate consistent, impactful, and versatile contributions to data science projects.

3 Results

3.1 RQ1: How the selected metrics provide information on the averages of data scientists' individual contributions ?

The following charts present the boxplot for each of the metrics. The number of commits is a key indicator of developer productivity. Figure 2 shows that most developers have a median of more than 400 commits. There are a few outliers with significantly higher numbers of commits, with one developer having over 2,000 commits in the last two years. The number of files changed is another important metric. The analyses shows the median number of files changed is around 5000. There are extreme cases where some developers have changed over 20,000 files, indicating a wide variation in contribution levels. Figure 2 also presents the median number of LOC added is less than 100K. There are notable outliers, with some developers adding over five million lines of code, which illustrates the substantial contribution of a few developers.

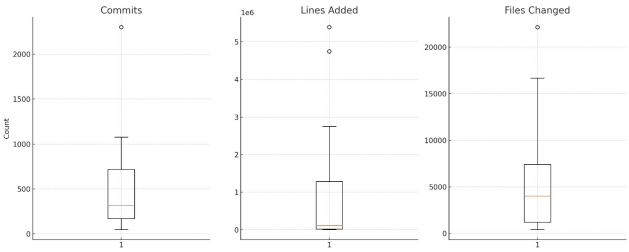

Figure 2: Data Scientists Metrics

Due to the variance of the outliers shown in the data scientist data, our analysis on the top-10 data scientists based on high averages in the 3 metrics presented. This strategy allows us for a detailed examination of the most influential contributors. We have performed an exhaustive analysis to distinguish these specialists based on their proficiency in these metrics.

Figure 3 shows the commit distribution by individual data scientists in their projects over the last 2 years. To maintain the developers anonymous, we utilize "DS" followed by a number. With 2,299 commits, DS1 is the most prolific project, while DS10 has the fewest, with only 256 commits. This analysis of commit patterns offers insights into the dynamics of development and individual contributions within these projects.

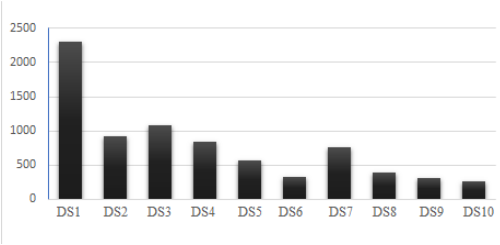

Figure 3: Number of Commits by Data Scientists

Figure 4 shows the LOC added by each data scientist to their projects in the last 2 years. Notably, DS5 contributed over 5.3 million LOC, indicating significant impact, while DS2 added 104,057 lines. This metric quantitatively assesses code contributions, highlighting the varying scales of individual efforts. The contrast between the highest and lowest contributors underscores the diversity in engagement and output among data scientists, providing a nuanced perspective on coding efforts within the analyzed projects.

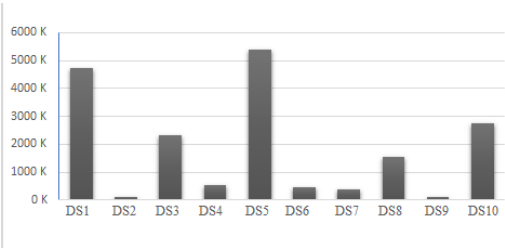

Figure 4: Lines of Code Added by Data Scientists

Figure 5 presents the analysis of 2-year contributor activity of the developers, showing the number of files modified by each data scientist. DS1 had the most extensive impact, modifying 22,134 files, while DS9 had the least impact, altering 1,608 files. This figure details the scope of individual contributions, highlighting the degree to which each data scientist has impacted the project through file modifications.

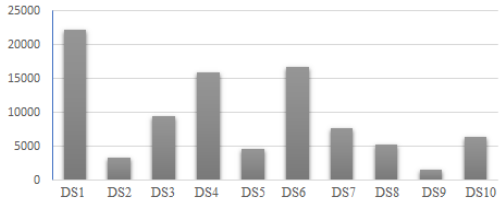

Figure 5: Number of Files Modified by Data Scientists

3.2 What are the characteristics of individuals identified as experts in data science projects?

A total of 69 data scientists were subjected to analysis. To classify them, certain categories were established. A *Progressive Data Scientist* is characterized by a preference for adding code to projects, with at least 60% of their activity spent on additions and no more than 40% on deletions. This group represents 53% of the analyzed data scientists. The *Conservative Data Scientist* classification identifies those who contribute most to projects by removing code, with at least 60% of their activity focused on deletions and no more than 40% on additions, comprising 22% of the total. *Standard Data Scientists* represent a balanced approach, aiming to maintain equilibrium between code additions and removals, with neither exceeding 60% of their total activity; they make up 25% of the group.

Furthermore, the classification extends to language usage. *Mono-Language Data Scientists* predominantly use a single programming language, with no more than 40% of their work performed in a secondary language, accounting for 68% of the data scientists. Conversely, *Multi-Language Data Scientists* contribute in multiple programming languages, with at least 40% of their lines of code dedicated to a secondary language in addition to their primary language, representing 32% of the analyzed data scientists.

The majority of data scientists in our sample demonstrated a progressive coding behavior (53%), followed by those who adhered to a standard approach (25%). A smaller proportion of data scientists were identified as conservative in their coding practices (22%). In terms of programming language usage, a significant majority of data scientists (68%) predominantly contributed using a single programming language (Mono-language). Conversely, 32% of the data scientists were classified as multilingual, indicating proficiency in contributing to projects using multiple programming languages. These findings offer insights into the coding behavior of data scientists and can inform project management strategies aimed at optimizing coding contributions within data science projects.

We identified that data scientists who consistently score high in at least 2 of 3 specific metrics qualify as experts. Recognizing that expertise extends beyond technical skills to broader engagement, we pinpointed developers with a unique combination of skills, commitment, and influence by focusing on these metrics. Table 2 lists the five most active and collaborative data scientists based on specified metrics. The first column identifies these top-5 Data Scientists (DS), showing their activity across each metric. The second and third columns detail the total lines of code added and deleted by each DS, reflecting their involvement in code optimization or removal of redundant segments. The fourth column shows the total files modified, indicating contribu-

tions to various software components, while the fifth column records their total commits. Their sustained, active participation marks them as influential data science experts, highlighting their significant impact on project outcomes.

Based on their contributions to the project, DS1 emerges as a significant contributor, having added 4,743,548 lines of code (LOC) and deleted 5,506,873 LOC. DS1 has modified 22,134 files and made 2,299 commits, which are among the highest numbers in all metrics. This can be associated with a change to the entire project code or a automatically generated code. DS3 showcases a noteworthy balance between the number of added and deleted LOC (522,918 and 395,256, respectively) and a considerable number of changed files (15,923) and commits (834). DS2 and DS5 have moderate levels of involvement in their project, with varying degrees of contributions across different metrics. For example, DS3 has added 522,918 LOC, the lowest value for LOC, but changed 15,923 files which is the second highest value, in addition to have made 834 commits. These variations emphasize the diverse roles and contributions of each data scientist within the project, underscoring the complexity of data science roles within team environments.

Table 2: Overview Experts in Data Science

DS	Added LOC	Del LOC	#Changed Files	#Commits
DS1	4,743,548	5,506,873	22,134	2,299
DS2	2,341,474	4,04,032	9,520	1,078
DS3	522,918	395,256	15,923	834
DS4	5,387,978	4,001,774	4,618	568
DS5	2,750,528	270,874	6,338	256

3.3 Which programming languages are mostly used by data science experts?

Our analysis of programming languages used by data science experts provides a comprehensive view of their preferences and trends. Unlike general examinations of languages used by traditional software developers [12], our focus is on the number of files modified by each data scientist within data science projects. This detailed analysis reveals the linguistic preferences crucial to proficiency in data science. Additionally, Python emerges as the primary language, with significant proficiency also noted in TypeScript and Rust.

Figure 6 illustrates the language proficiency of 5 selected data science experts, showcasing the programming languages each expert commits to. This analysis highlights the diverse skill sets of each developer and is crucial for understanding expertise distribution. It provides insights into the team's collective skills and areas of specialization, serving as a reference for the language foundation of a data science project. Notably, DS4 modified the most files per language, with 2,714 in TypeScript, 1,083 in Rust, and 164 in JSON, indicating substantial involvement and versatility across various programming languages.

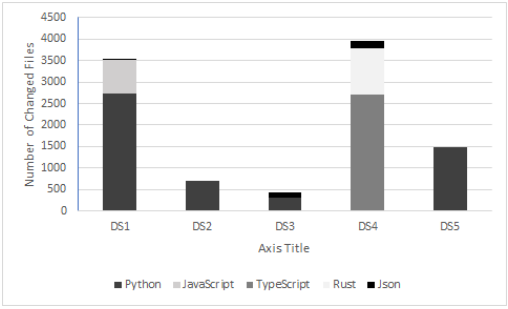

Figure 6: Overview Languages per Experts

4 Threats to Validity

The study presented in this paper has some limitations that could potentially threaten our results, as we explain next.

Construct Validity. Although quantifying data science proficiency through GitHub activities might not adequately represent its complexity, we addressed this by choosing a variety of indicators to give a broader picture. This decision is aligned with related literature [7].

Internal Validity. We used a multifaceted selection strategy that took into account project activity levels, diversity, and popularity metrics like stars to mitigate selection bias of GitHub projects. The goal of this strategy was to provide a sample of projects that was more representative.

Conclusion Validity. In our study, although the selected metrics may have led us to wrong conclusions, we chose them based on related literature [7]. Moreover, cross-discussions among the paper authors often took place until we reached a common agreement about our main findings.

External Validity. Given the small sample size, we have exercised caution when extrapolating our results. To improve the generalization of our findings, future research may expand on this study by involving a larger number of contributors and projects.

5 Related Work

Kim et al. [5] examined the changing role of data scientists in software development using Microsoft as a case study. They interviewed 16 data scientists from different Microsoft product groups to learn about their jobs, educational backgrounds, and working methods. The authors identified five working styles: Polymaths, Team Leaders,

Insight Providers, and Modeling Specialists [5]. While Kim et al. [5] study focused on understanding the roles and techniques of data scientists in teams, our research aims to recognize and measure data science proficiency among developers by analyzing software repositories. We offer a quantitative analysis of data science contributions within software repositories, complementing Kim et al.'s qualitative perspective. These studies deepen our understanding of the evolving landscape of software engineering, where data science expertise significantly impacts project dynamics and outcomes.

Saltz et al. [8] investigated software engineers transitioning to data engineering roles using a case study at a big data consulting firm. They found significant differences in the skills and flexibility required for software engineering and data engineering. While Saltz et al. [8] concluded that not all software engineers have the necessary skills for data engineering roles, our study provides further insight into the specific skills and attributes prevalent among data science professionals.

By identifying and quantifying these characteristics, our research contributes to understanding the diverse skill requirements within the data science domain, informing decisions related to role transitions and development strategies. These studies provide valuable insights into software engineering, highlighting the multiple pathways and skill sets shaping career trajectories in the field.

Oliveira et al. [7] presented a study on the efficacy of two source code analysis models (Changed Files and Changed Lines of Code) in detecting programming talents. They analyzed 110 GitHub developers to evaluate the accuracy of these models in detecting hard skills, back-end and front-end profiles, and testing. The study found that although these models show promise, their accuracy is somewhat poor, indicating that automated skill assessment techniques need improvement [7].

While related work has investigated software engineers' technical skills, these studies did not explore data science elements in the contexts examined in this paper. Our research focuses on using GitHub code metrics, such as commits and changed files, to identify data science experts.

6 Conclusion

In this paper, we conducted a study with 18 data science projects using mining software repositories to identify data science experts. We analyzed specific metrics and focused on 69 developers for deeper analysis, categorizing the main contributors based on these metrics. This enabled a focused examination of influential data specialists and their roles in advancing data science projects. We also created profiles based on selected metrics (LOC, Number of Commits, and Changed Files) and the programming languages used.

Future research could extend this study by incorporating a larger and more diverse dataset across various domains and industries, helping to validate and generalize our findings. Adding new metrics, such as code complexity and review feedback, could deepen our understanding of expertise in data science repositories. Additionally, examining longitudinal trends could reveal how expertise evolves and impacts project outcomes.

References

[1] A. Begel and T. Zimmermann. Analyze this! 145 questions for data scientists in software engineering. In *Proceedings of the 36th International Conference on Software Engineering*, pages 12–23, 2014.

[2] Christof Ebert, Jens Heidrich, Silverio Martínez-Fernández, and Adam Trendowicz. Data science: Technologies for better software. *IEEE Software*, 36:66–72, 2019. URL `https://api.semanticscholar.org/CorpusID:204863495`.

[3] A. E. Hassan and T. Xie. Software intelligence: The future of mining software engineering data. In *Proceedings of the FSE/SDP Workshop on the Future of Software Engineering Research, FoSER 2010*, pages 161–165, 2010.

[4] A. Jordan and D. Berleant. Data science knowledge and skills that reliability engineers need: A survey. In *2023 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–6, 2023.

[5] M. Kim, T. Zimmermann, R. DeLine, and A. Begel. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering*, pages 96–107, 2016.

[6] S. Kourtzanidis, A. Chatzigeorgiou, and A. Ampatzoglou. Reposkillminer: Identifying software expertise from github repositories using natural language processing. In *35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1353–1357, 2020.

[7] J. Oliveira, M. Souza, M. Flauzino, R. Durelli, and E. Figueiredo. Can source code analysis indicate programming skills? a survey with developers. In *Quality of Information and Communications Technology*, pages 156–171, 2022.

[8] J. S. Saltz, S. Yilmazel, and O. Yilmazel. Not all software engineers can become good data engineers. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2896–2901, 2016.

[9] A. Santos, M. Souza, J. Oliveira, and E. Figueiredo. Mining software repositories to identify library experts. In *Proceedings of the VII Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS '18)*, pages 83–91, New York, NY, USA, 2018. Association for Computing Machinery.

[10] T. Siddiqui and A. Ahmad. *Data mining tools and techniques for mining software repositories: A systematic review*, pages 717–726. Springer, 2018.

[11] D. Spadini, M. Aniche, and A. Bacchelli. Pydriller: Python framework for mining software repositories. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2018*, pages 908–911, 2018.

[12] TIOBE. Tiobe index for april 2024, 2024.

[13] Alexander Trautsch, Steffen Herbold, and Jens Grabowski. Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2020.

[14] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer Science & Business Media, 2012.