ECS 260 Project Progress Report

Professor: Vladimir Filkov

Working Group: Bobby Missirian

Karamjeet Singh Gulati

Pu Sun

UC DAVIS

UNIVERSITY OF CALIFORNIA

March 19, 2024

1 Introduction

In recent years, the Open Source Software (OSS) movement has revolutionized the way software is developed, distributed, and maintained. OSS projects have become integral to technological innovation, driving the development of key software components used in various industries. However, despite the growing prominence of OSS, understanding the factors that contribute to the success and sustainability of these projects remains a challenge. This research aims to address this gap by focusing on the incubation phase of OSS projects, a critical period that can determine their long-term viability. The Apache Software Foundation Incubator (ASFI) serves as a nurturing ground for early-stage OSS projects, providing them with the necessary support and guidance to evolve into sustainable, community-driven endeavors. This research centers on the ASFI, investigating the dynamics of OSS projects during their incubation to uncover the key determinants of success. The code repository for this research can be found at https://github.com/ksg98/ECS260-Software_engineering.

The specific research questions guiding this study are:

Initial Activity Patterns: *Can the patterns of activity in the early stages of a project predict its long-term success?*
Diversity of Contributions: *Does the diversity of contributions, in terms of both the number and nature of contributions, correlate with the sustainability of the project?*
Communication Patterns: *How do the patterns of communication among project contributors affect the outcome of the project?*

The motivation for this project is driven by the need to understand the critical factors that influence the success of OSS projects. By focusing on the incubation phase of projects in ASFI, this research seeks to provide valuable insights for project initiators, contributors, and stakeholders on how to enhance the prospects of OSS projects. Furthermore, the findings of this study aim to contribute to the broader knowledge base on OSS development, offering guidance for future projects and fostering a more vibrant OSS ecosystem.

The contributions of this project are multifaceted:

Empirical Analysis: Conducting a comprehensive analysis of OSS projects incubated at the ASFI, this research provides empirical evidence on the relationship between early project activities, contribution diversity, communication patterns, and project success.
Theoretical Insights: By integrating concepts from OSS sustainability, community dynamics, and structural contingency theory, this study offers theoretical insights into the complex interplay between project organization and socio-technical interactions in the context of OSS development.
Practical Recommendations: The findings of this research are expected to yield practical recommendations for OSS project initiators and contributors, guiding them in strategies to enhance project success and sustainability during the critical incubation phase.
Methodological Advancements: Utilizing advanced statistical modeling and data analysis techniques, this study contributes to the methodological approaches used in the study of OSS projects, paving the way for future research in this area.

Overall, this research aims to deepen our understanding of the dynamics that underpin the success of OSS projects, with a particular focus on the crucial incubation phase, ultimately contributing to

the advancement of the OSS movement.

2 Background

Our study explores the key factors influencing the success of Open Source Software (OSS) projects during their incubation at the Apache Software Foundation Incubator (ASFI), focusing on early activities, contribution diversity, and communication patterns. We aim to understand how these elements impact sustainability and success, offering insights for enhancing OSS project strategies. The theoretical foundation integrates concepts from OSS sustainability and the role of community dynamics to emphasize the nuanced interplay between project organization and socio-technical interactions.

Previous research has highlighted the importance of contributions from both minor and major contributors in OSS projects for sustainability, underscoring the need for a diverse range of contributions [1]. Similarly, the impact of early participation factors on OSS project sustainability has been stressed, underscoring the significance of fostering a vibrant community in the initial stages of development [2]. The inefficiencies associated with forking-based development in OSS have been identified, with modularity, centralized management, and handling hard forks identified as crucial factors for project success [3]. ASF's mature governance and support in propelling projects towards sustainability highlight its role in nurturing early-stage projects [4]. Additionally, investigating forking and pull request practices within OSS communities has yielded insights into best practices for maintainers to enhance contribution efficiency and foster community cohesion [5].

Our approach to exploring OSS project sustainability, particularly within ASFI, is further informed by the ASF Incubator dataset, which provides an in-depth look at open-source software development, featuring data on contributors, project statuses, code changes, and communications [6]. This comprehensive dataset aids in understanding the dynamics of OSS project development and success factors, offering a unique perspective on the incubation process and the factors that contribute to the sustainability of OSS projects.

Our work fits into the existing research on Open Source Software (OSS) projects by specifically addressing the incubation phase within the Apache Software Foundation Incubator (ASFI), a critical yet underexplored period in OSS project development. While prior studies have explored various facets of OSS projects, our research concentrates on the early activities, diversity of contributions, and communication patterns that characterize this initial phase. By investigating these elements, we aim to uncover the key factors that contribute to the success and sustainability of OSS projects. Our study expands the understanding of OSS project dynamics by analyzing the impact of initial activities and diverse contributions on project outcomes. Additionally, we examine how communication patterns within these projects influence their graduation or retirement from the incubator. This research provides valuable insights for OSS project strategists, offering guidance on enhancing project success and sustainability during the pivotal incubation phase.

3 Data

The Apache Software Foundation Incubator (ASFI) dataset provides a panoramic view of the development processes and community dynamics and aids in discerning the factors that contribute to the success or failure of OSS projects. The ASFI dataset is crucial for analyzing OSS development, offering insights into processes, community dynamics, and project success factors. It covers contributor details, code changes, and communications, providing a comprehensive view of ASF projects.[6]. It includes:

People Dataset: Contains contributor details such as identities, roles (e.g., developer), and associated emails.

Lists Dataset: Includes project start/end dates, aiding in tracking incubation periods and categorizing outcomes (graduated, retired, incubating).

Aliases Dataset: Maps multiple aliases or emails to a single contributor identity, ensuring accurate identification.

Filelist Dataset: Tracks file changes, revealing project evolution and providing insights into development practices and scope changes.

Commits Dataset: Similar to Filelist but focuses on changes to files, offering insights into project adaptability and maintenance practices.

Messages Dataset: Captures email communications between contributors, valuable for studying communication patterns and decision-making processes.

This dataset provides an unparalleled resource for exploring the multifaceted nature of OSS development within the ASF ecosystem. By covering every aspect from contributor identities to code changes and communications, this dataset enables a comprehensive analysis of OSS projects

4 Methodology

4.1 RQ1

This comprehensive methodology lays out a strategic approach that spans from initial data gathering to sophisticated statistical modeling, aiming to identify the early indicators pivotal for a project's sustainability and success. We begin our analytical journey by examining our study design.

4.1.1 Study Design

Our study adopts a precise longitudinal quantitative research design, focusing specifically on the early activity patterns within the Apache Software Foundation Incubator (ASFI) and their influence on the long-term success of open-source software (OSS) projects. By analyzing 269 projects, each marked by their eventual success ("graduated") or failure ("retired"), we aim to unearth the critical indicators present in the initial stages of project development. This design is meticulously chosen to capture the essence of project activities and interactions early on, providing insights into how these initial engagements correlate with project outcomes.

The heart of our study design lies in its ability to meticulously track and analyze the temporal dynamics of project development and communication within the ASFI. Focusing on the crucial first three months post-project initiation, we delve into quantifying team engagements, contributions, and communication patterns. This approach not only highlights key predictors of success in the OSS environment but also offers a deeper understanding of the incubation process, shedding light on the factors that contribute to the sustainability and triumph of OSS projects.

4.1.2 Variables Considered

Dependent Variable

Project Outcome: This binary variable categorizes projects as either 'graduated' (1) or 'retired' (0), reflecting their success or failure, respectively.

Independent Variables

Incubation Time: The duration (in months) a project spends in the incubation stage.

Number of Commits: Total commits made during a project's lifecycle, indicating development activity.

Number of Committers: Count of unique individuals contributing commits, reflecting community engagement.

Number of Emails: Total emails exchanged, measuring communication intensity among developers.

Number of Email Senders: Count of distinct individuals initiating email communications, signifying community participation depth.

Number of Files: The scale of a project's codebase, inferred from the total file count within the project repository.

4.1.3 Data Collection

An intricate data collection process was employed, harnessing data from the *commits, filelist, lists* and *messages* datasets. This process was underpinned by a rigorous methodological framework aimed at ensuring the reliability and validity of the analysis. Data extraction involved filtering relevant information from the datasets, focusing on metrics pertinent to the initial three months post-project start. The datasets underwent rigorous cleaning to address inconsistencies, duplicates, and missing values, ensuring data integrity for analysis. Additionally, fields related to dates, including commit-datetime, start-date, and end-date, were converted to datetime objects to facilitate temporal analysis.

4.1.4 Outline Analysis Methods

This approach to data preprocessing and initial activity analysis is both thorough and deliberate. By preparing the data and scrutinizing early project activities, we aim to glean insights into the project dynamics during the critical initial months. This analysis is crucial for our next steps, which involve correlating these activities with project outcomes.

Data Preprocessing: During data preprocessing, data from the commits, filelist, and lists datasets were loaded into Pandas DataFrames for manipulation. To ensure consistency, uniformity in email addresses and developer identifiers was established. Missing values were managed through various methods, such as imputation, exclusion, or alternative approaches, depending on the context and impact on analysis.

Initial Activity Analysis: A crucial window for initial project activity was identified as the first three months following each project's start date. Key metrics within this window—total commits, unique contributors, files added, and files modified—were calculated to assess the distribution and variability of initial project activities.

Correlation with Project Outcomes: Initial activity data was merged with final project outcomes using listid to classify projects accordingly. Logistic regression analysis was employed to explore the relationship between initial activities and project outcomes, with interaction terms considered for deeper insights into the potential interactions between different variables.

Statistical Analysis and Modeling: The logistic regression model's performance was evaluated using accuracy, precision, recall, and F1-score metrics, derived from a classification report and confusion matrix. Multicollinearity was checked using the Variance Inflation Factor (VIF), with adjustments made to improve the model's representation of observed data. To ensure the model's adequacy in representing observed data, a Hosmer-Lemeshow test was conducted.

Advanced Analysis Stages: Residual analysis was conducted to identify any violations of the logistic regression model's assumptions. Longitudinal data analysis techniques were applied to track

project evolution over time. Advanced statistical models, including Random Forests, Support Vector Machines (SVM), and Neural Networks, were explored for their ability to capture complex relationships within the data. Model refinement strategies were implemented based on insights from performance evaluation and residual analysis.

Through this targeted analysis, we aim to unveil critical early indicators, enriching our understanding of the factors that propel OSS projects towards successful graduation within the ASFI framework.

4.2 RQ 2

4.2.1 Study Design

We analyze commit patterns in open-source software (OSS) projects to assess their impact on project success. Our approach is structured around data collection, preprocessing, and detailed analysis, leveraging the available data to explore the dynamics of code contributions and their correlation with project outcomes.

Data Preprocessing Initial steps involve cleaning the data for consistency, removing duplicates, and addressing missing values. Timestamps are normalized to ensure accurate temporal analysis. Specifically, we remove outliers based on project start and end times for accurate commit analysis, while correcting for over-counting in multi-file commits to understand unique development efforts.

Temporal Analysis of Commits We conduct a temporal analysis to explore how commit patterns evolve throughout the lifecycle of OSS projects. This involves aggregating communications by bi-monthly intervals and comparing patterns across projects of different outcomes. This involves understanding the distribution of projects across our target time frame, so we classify projects by status and visualize their active durations.

Quantitative Analysis of Commit Volume The volume of commits is quantified and analyzed in relation to project outcomes. We create box plots for commit volumes and operation-specific line change counts, contrasting graduated and retired projects. We hypothesize that successful projects may exhibit different communication volumes or patterns compared to unsuccessful ones.

Statistical Testing Tried and true statistical methods, such as regression analysis and box plots, are employed to visualize communication patterns and recognize their impact on project outcomes. We analyze commit activity patterns over time using bimonthly bins for ongoing, graduated, and retired projects. This step is crucial for validating our hypothesis.

Predictive Modeling Leveraging machine learning techniques, we aim to predict project outcomes based on commit patterns. This involves training models on a subset of the data and testing their predictive power on unseen data. Our goal is to isolate the most important variables that indicate project success and failure.

4.2.2 RQ 2: Data Collection

There are two main data tables of importance with a few key columns used in our commit analysis.
Commits Table

list An identifier linking to the specific project mailing list corresponding to the same one in the lists dataset.

messageid A unique identifier for each commit message. In some case, there are many duplicate commits with the same message id that are really one commit.

commit_datetime The datetime of the commit.

file_operation The operation performed on the file (e.g., add, modify).

file_name The name of the file involved in the commit.

addlines The number of lines added in the commit.

dellines The number of lines deleted in the commit.

Lists

listid A unique identifier for each mailing list.

listname The name of the mailing list. In practice, this is same name as the project.

pj_alias Project alias associated with the mailing list.

status Indicates the status of the project mailing list. In other words, has this project graduated, been retired, or is it ongoing?

start_date The start date of the mailing list.

start_date The end date of the mailing list, with some missing values due to ongoing projects.

4.2.3 RQ 2: Variables Considered

From the list table, listid, pj_alias, and status will help correlate the contributions with project outcomes (graduated, retired), providing a comprehensive view of how diverse contributions influence project sustainability. Further, the lifespan date range, derived from the start_date and end_date in the list table, will be utilized for regularizing commit contributions. This data forms the backbone of our detailed analysis of commit patterns.

The commits table provides data on file operations (additions, modifications, deletions) and the diversity of the affected files. The temporal element of the columns in the Commits table shows the frequency of these operations over time. Data features from the commits table, including file_operation, addlines, and dellines, are instrumental in assessing the diversity of contributions. These features enable us to analyze the types and extent of changes made to project files, offering insights into the variety of contributions. Further, we conduct a regression analysis on bimonthly binned commit data to determine the average commits over time, commit consistency, and the trend direction. We use the r-value of our regression analysis as heuristic for consistency since it goes down when there is greater deviation from the average linear trend. Finally, we use SHAP analysis to know for sure which factors are predictive of project outcome.

4.2.4 RQ 2: Analysis Methods

The methodology for analyzing commit patterns in OSS projects integrates various analytical techniques to understand their impact on project success. Data preprocessing ensures the dataset's reliability, setting a solid foundation for subsequent analysis. Temporal analysis uncovers the evolution of commit patterns over project lifespans, offering insights into project activity periods. Quantitative analysis and statistical testing, which employ regression analysis and box plots, identify trends and differences in commit patterns between successful and unsuccessful projects.

Leveraging advanced algorithms to forecast project success based on historical commit patterns allows for the identification of key predictive variables. A variety of competing machine learning models are trained on factors including commit volume, consistency, and the nature of changes (additions, deletions). This approach not only predicts outcomes but also isolates factors most indicative of project health, providing actionable insights for project management. We settle on Gradient Boosting as our best model for the dataset and conduct a SHAP analysis.

Gradient boosting is particularly suited for the analysis of OSS project data, which is plagued by imbalanced classes (e.g., more retired projects than graduated ones). This machine learning technique iteratively corrects errors from previous models, focusing on difficult-to-predict instances, making it effective for datasets where success cases might be rare. By emphasizing these harder cases, gradient boosting can improve prediction accuracy in imbalanced contexts, ensuring that the minority class

(e.g., successful projects) is adequately represented and learned from, leading to more nuanced insights and predictions.

SHAP (SHapley Additive exPlanations) analysis is utilized to interpret gradient boosting results, particularly useful for complex models. It provides insights into how each feature contributes to the model's prediction for a given observation, distinguishing critical factors influencing project outcomes. Essentially, Shapley values distribute the total gain of a coalition (e.g., model prediction over a dataset) fairly among its players (features) based on their contribution. Mathematically, it involves averaging the marginal contributions of a feature across all possible feature permutations. SHAP values reveal the impact of each feature on the gradient boosting model's prediction.

4.3 RQ 3

4.3.1 Study design

Our study investigates the impact of initial communication patterns on ASFI project success, examining 269 OSS projects during the critical first three months after launch and categorizing them by their eventual graduation or retirement. Rigorous data cleaning is performed to ensure quality, while advanced NLP techniques, including TF-IDF vectorization and K-means clustering, classify communications into meaningful parts such as implementations, tools, data management, frameworks, and services. We integrate bifurcation identification to explain the evolution of projects and employ a random forest model, supplemented by SHAP value analysis, to establish predictive correlations of communication types. Our analysis aims to reveal early indicators of project success, providing insights into the underlying dynamics that predict the long-term sustainability of open source software development.

4.3.2 Data Collection

Data Cleansing: We performed thorough data cleaning to remove any duplicates, null values, and irrelevant information, ensuring a pristine dataset for analysis.

NLP Classification: Textual data were transformed into a matrix of TF-IDF features using TfidfVectorizer, emphasizing term significance. K-means clustering algorithm then categorized communications into distinct classifications reflective of the project's discourse: Implementation and Specifications, Tools and Platforms, Data Management and Processing, Frameworks and Systems, Services and Integration

Fork Data Identification: Essential to understanding collaborative dynamics, fork data were identified and marked to trace the evolution and branching of projects.

4.3.3 Variables Considered

In addition to the data collection phase, our analysis focuses on specific variables in the "commit" and "list" datasets to reveal the impact of early communication patterns on project outcomes within the Apache Software Foundation incubator. These variables are critical to our investigation and include:Project ID and Name: Crucial for linking submission activities and discussions to their respective projects, ensuring accurate and consistent data analysis across various data sets; General Discussion: Aspects of communication extracted from the project mailing list were analyzed to measure community engagement and the collaborative climate of the project; Intro: Initial communication is critical to setting project goals and early team dynamics, and we carefully examine its potential impact on project outcomes; Commit Data ('ref or sha'): Commit metadata provides insights into patterns of development activity, including the frequency and nature of contributions, which indicates the

health and momentum of the project; Start Date and Date ('start date', 'date'): These time variables help put the data into context, allowing for longitudinal analysis of communication and development activities related to the project timeline; Project Outcome ("Status"): same as above RQs, serves as the dependent variable, allowing us to describe the factors that led to the success or termination of the project.

4.3.4 Analysis Methods

High data processing and time series analysis In this initial phase, we reorganized, cleaned, and standardized communication data from ASF incubator projects. We then used time series analysis to dissect communication trends over time to gain a detailed understanding of the evolution of communication across different project phases. This approach helps identify patterns and anomalies in the frequency and content of communications.

Comprehensive descriptive analysis We extend the descriptive analysis, which involves not only calculating average communication messages, but also exploring their distribution, variability and temporal changes. By integrating time series data, we aim to enhance our understanding of communication dynamics in project development by revealing underlying patterns and laying the foundation for other statistical analyses.

Integration of statistical testing and machine learning Use analysis of variance to identify significant differences in communication between project statuses. And a random forest model is adopted to identify and rank the importance of various communication features. Subsequently, SHAP values are calculated to interpret the random forest results, providing insight into feature contributions and their impact on model predictions. This dual approach provides a nuanced understanding of the factors that influence project success.

Diverse correlation and regression analysis It combines a modularity index and an expanded correlation matrix to evaluate the relationship between communication indicators and project outcomes. In the regression analysis, we not only focus on reconciling message counts but also incorporate important variables identified through machine learning analysis. This approach aims to build a predictive model that accurately represents the impact of different communication aspects on project success, providing actionable insights for an effective project communication strategy.

Comparative analysis of traffic volume Incorporating a comparative analysis, we examined communication volume before and after important project milestones. This comparison aims to identify changes in communication patterns, providing insights into how different project phases impact team interactions. Understanding these shifts can provide valuable clues about communication needs and challenges at various project stages.

Variance Inflation Factor (VIF) Analysis Reinforcement The VIF analysis remains crucial in verifying the regression model's reliability. It now includes an examination of multicollinearity among a broader set of predictors, ensuring the enhanced model's robustness and the validity of our conclusions regarding communication's impact on project success within the ASF incubator.

5 Result

5.1 RQ 1

The analysis reveals that successful projects often have higher median values for total commits and unique contributors, indicating active early development and diverse involvement. In contrast, retired projects show lower medians in these metrics, suggesting less initial activity. Graduated projects also exhibit higher medians for files added and modified, indicating a more dynamic and evolving initial

codebase compared to retired projects. While robust initial activity generally correlates with project graduation, outliers indicate that this is not always the case, as shown in Figure 1.

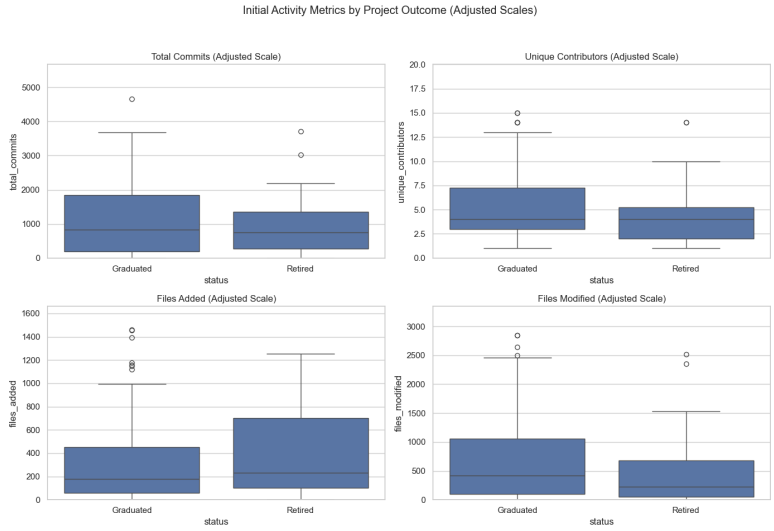

Figure 1: Analysis of Initial Activity Metrics by Project Outcome

The analysis commenced with the evaluation of a logistic regression model, aiming to assess the influence of key activity metrics on project success. It was found that the total number of commits had a weak positive correlation with successful project graduation, with a correlation coefficient of 0.0698. However, this metric did not show a statistically significant relationship with project outcomes (p-value = 0.3807). Similarly, the number of unique contributors, a measure of community engagement, also demonstrated a weak positive correlation with graduation (correlation coefficient = 0.1071), but did not reach statistical significance (p-value = 0.1777).

Regarding codebase activity, additional insights were gleaned. The number of files added to a project presented a weak negative correlation with successful graduation (correlation coefficient = -0.1025, p-value = 0.1971), suggesting that simply adding files to a project is not a reliable indicator of success. On the other hand, the number of files modified showed a weak positive correlation (correlation coefficient = 0.1146), with a p-value of 0.1491, indicating that active development work, rather than the volume of files, may be more indicative of project health and trajectory.

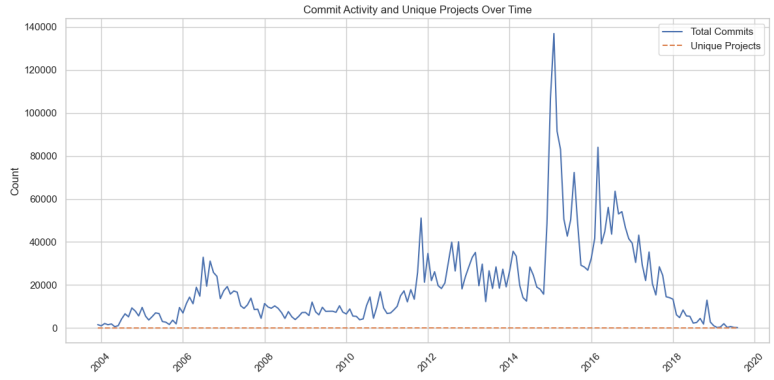

Figure 2: The Commit Activity and Unique Projects Over Time

Figure 2. illustrates the fluctuations in commit activity over time, alongside the constancy of

unique project initiations within the ASFI. Notably, there are spikes in commit volume, including one particularly sharp increase, which may correlate with special events like collaborative surges, hackathons, or significant project milestones. Despite these variations, the number of unique projects depicted remains consistent, indicating that while new projects continually enter the incubator, their developmental intensity differs. A noticeable downward trend in recent times points to a potential decline in activity, possibly due to migrations to newer platforms, project completions, or shifts in the open-source ecosystem. Additionally, any changes in the methodologies for data collection or reporting over time could contribute to the observed fluctuations in commit activity.

Model	Accuracy (%)	Precision (%)	Recall (%)
Logistic Regression	81	90	55
Random Forest Classifier	85	83	69
Support Vector Machine (SVM)	79	40	50
Neural Network Classifier	58	42	41

Table 1: Comparison of Model Performances: This table displays the accuracy, precision, and recall for each statistical model. The models exhibit varying levels of effectiveness in predicting the outcomes of projects within ASFI, with the Random Forest Classifier achieving the highest accuracy.

The analysis employed a Random Forest Classifier, which outperformed other models with an accuracy of 85% as shown in Table 1. This model underscored the hypothesis that active initial development and diverse community involvement are crucial for a project's success within ASFI. Additionally, Cook's distance measurements reassured that the model's predictions were not disproportionately influenced by any singular observation, lending confidence to the results.

The Support Vector Machine (SVM) and Neural Network Classifier were also tested but displayed limitations in their predictive abilities. The SVM showed a predisposition toward over-predicting the majority class, highlighting potential issues with class imbalance. The Neural Network model struggled with accuracy, likely due to overfitting, achieving an overall accuracy of 58

In summary, the results of the advanced statistical models suggest a complex interplay between early project activities and graduation outcomes. While higher initial commits, unique contributor counts, and extensive code modifications are generally associated with project success, they are not definitive predictors on their own. The insights derived from this analysis offer valuable guidance for strategizing robust initial development and engagement to enhance the likelihood of long-term project success within the ASF Incubator.

5.2 RQ 2

Preliminary analysis showed an order of magnitude higher averages and quartile ranges for bimonthly commits, commit slopes, and bimonthly addlines between graduated and retired projects. Although these metrics, seen in Table 2, are useful as decision making factors, they are unsuitable when used on their own for predicting project outcomes. Consequently, we compiled all relevant metrics related to the nature of a commit, including commit consistency, slope, bimonthly commits, bimonthly addlines, and total commits, into one boosting model with 90% accuracy. Then, we conducted a SHAP analysis and determined that the key discriminating variables are bimonthly commit count, commit consistency, and slope, respectively.

Overall, retired projects have an order of magnitude fewer addlines and bimonthly commits on average. They also have lower consistency on average and tend to have fewer commits over time. The

Feature	Statistic	Graduated	Retired
Addlines	25th percentile	39718.20	15887.23
	Median	122910.60	35002.93
	75th percentile	272014.37	84934.70
Commits	25th percentile	72.93	16.67
	Median	168.14	24.52
	75th percentile	346.21	52.72
Slope	25th percentile	-0.64	-5.94
	Median	9.62	-2.12
	75th percentile	39.85	-0.09

Table 2: The ranges of key features for graduated and retired projects.

most accurate model, a Gradient Boosting classifier with learning rate .1, found bimonthly commits, consistency, and commit trend to be the most important decision making factors by a wide margin.

Figure 3: SHAP values for the top 3 metrics, biannual commits (commits per active bin), commit consistency (r value), and commit trend direction (slope). Target is 1 for Graduated, 2 for Retired.

It is clear from Figure 3 that the distributions of slope and bimonthly commit count are substantially different. However, the degree to which the feature contributes to the prediction for a given project appears both highly nonlinear and dependent upon the project's other features. For this prediction task, complex models such as Gradient Boosting and Random Forest outperformed simpler models such as KNN, SVM, and MLP. Table 3 shows a how the features with highest SHAP score compare to other features, such as bimonthly addlines and total commits.

5.3 RQ 3

In a study of the ASF Incubator, we analyzed communication patterns across different project stages: graduation, incubation, and decommissioning. Graduation projects had 3997.88 general discussions and 1736.94 commit messages on average, while incubation projects recorded 4581.46 commit messages, and decommissioned projects had 3749.70 general discussions and 1166.28 commit messages. Despite these differences, ANOVA results indicated no significant variation in communication volume across phases, suggesting that communication quantity alone does not predict project success.

A negative correlation was observed between coordination messages and project success, particularly in the incubation phase, implying that increased coordination doesn't necessarily lead to success. The study also explored whether a higher ratio of commit messages to general discussions influences

Feature	SHAP value
Average Commits Per Active Bin	1.40
Commit Consistency	1.12
Slope	0.74
Total Commits	0.31
Addlines	0.16
Dellines	0.10

Table 3: The importance of each commit feature according to SHAP analysis of the Gradient Boosting predictor.

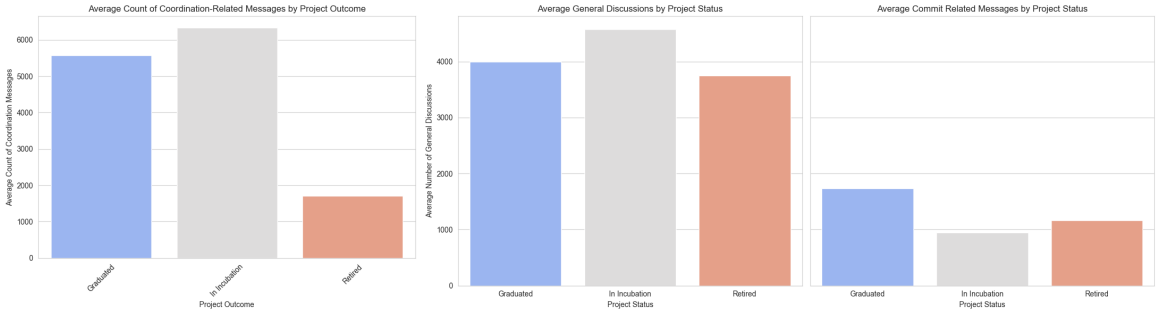

Figure 4: Average Number of General Discussions

project outcomes, along with the impact of different communication types (decision-making, information sharing, and problem solving) on project progression.

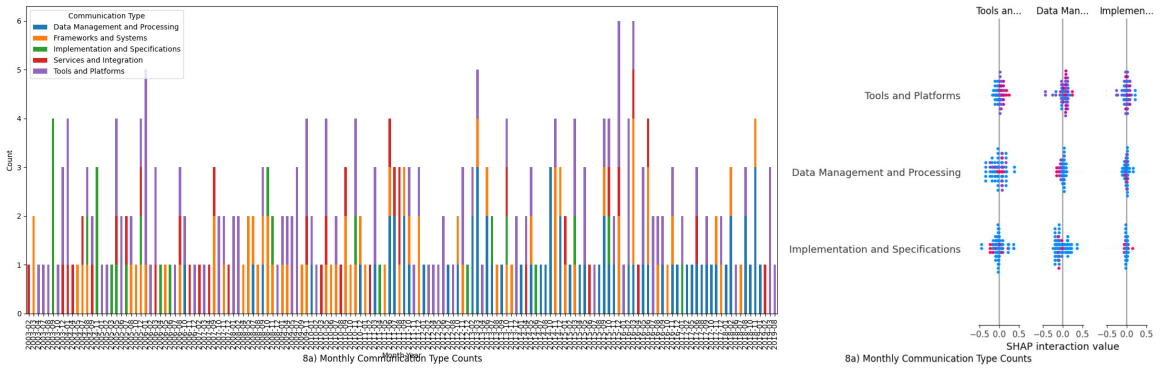

Figure 5: Monthly Communication Type analysis

Contrary to expectations, the volume of communication and project participation, as indicated by fork counts, showed inconsistent correlations. This underscores that the effect of communication on project success is complex and not merely dependent on the amount of interaction. Regression analyses further confirmed a significant negative relationship between coordination messages and project outcomes, suggesting that excessive coordination might be counterproductive.

A random forest classification model was used to predict project status with 71% accuracy, identifying "incubation," "graduated," and "retired" statuses. This model's feature importance scores emphasized the significant role of data management and processing communications in predicting project status. SHAP values provided additional insights as shown in figure 5, illustrating the im-

Feature	Feature Importance
Tools and Platforms	0.1360
Data Management and Processing	0.2799
Implementation and Specifications	0.2340
Frameworks and Systems	0.2271
Services and Integration	0.1229

Table 4: The importance of each communication type and its quantity classified by NLP to the random forest model in predicting the success of project incubation.

pact and importance of various communication types on the model's predictions, offering a deeper understanding of communication dynamics within the ASF Incubator.

6 Discussion

The key factors influencing the success of Open Source Software (OSS) projects during their incubation at the Apache Software Foundation Incubator (ASFI), the main findings highlight the importance of initial activity patterns, diversity of contributions, and communication patterns. The analysis reveals that successful projects tend to have higher median values for total commits and unique contributors, indicating active early development and diverse involvement. However, the presence of outliers suggests that robust initial activity is not a definitive predictor of success. Additionally, the study found that graduated projects have significantly higher median and interquartile commit counts compared to retired projects, with key variables for project success identified as bimonthly commit count, commit consistency, and commit trend direction. The research also uncovers that the quantity of communication alone does not predict project success, as there are no significant changes in the amount of communication across different project phases. However, there is a negative correlation between coordination messages and project success, particularly in the incubation stage.

6.1 Strengths and Limitations

The research benefits from the comprehensive dataset provided by the ASFI and the use of advanced statistical models, such as Random Forest and Gradient Boosting, to analyze the data. However, the focus solely on projects within the ASFI may limit the generalizability of the findings. Moreover, some metrics, such as the number of commits and unique contributors, show weak correlations with project success, indicating that additional factors may need to be considered.

6.2 Future work

Future efforts in this area could include expanding the analysis to other OSS ecosystems to validate the findings and explore additional factors influencing project success. A longitudinal analysis of projects from inception to completion could provide deeper insights into the dynamics of OSS project development and success. Additionally, incorporating qualitative research methods could enrich the understanding of the social and organizational aspects influencing OSS project outcomes. Overall, the research provides valuable insights into the factors contributing to the success of OSS projects during their incubation phase, emphasizing the importance of early activity patterns, diversity of contributions, and communication patterns. Further research is needed to fully understand the complex dynamics of OSS project development and success.

7 Team Membership and Attestation

Team members Karamjeet Singh Gulati, Bobby Missirian, Pu Sun participated sufficiently.

References

[1] undefinedtefan Stănciulescu, Likang Yin, and Vladimir Filkov. Code, quality, and process metrics in graduated and retired asfi projects. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, page 495–506, New York, NY, USA, 2022. Association for Computing Machinery.

[2] Wenxin Xiao, Hao He, Weiwei Xu, Yuxia Zhang, and Minghui Zhou. How early participation determines long-term sustained activity in github projects?, 2023.

[3] Anurag Dhasmana, Arindaam Roy, Divjeet Singh Jas, Kiranpreet Kaur, and Pinn Prugsanapan. Forking around: Correlation of forking practices with the success of a project, 2021.

[4] Likang Yin, Zhuangzhi Chen, Qi Xuan, and Vladimir Filkov. Sustainability forecasting for apache incubator projects. In *Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '21)*, page 12, New York, NY, USA, 2021. ACM.

[5] Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. What the fork: A study of inefficient and efficient forking practices in social coding. In *Proceedings of the 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '19)*, page 12, New York, NY, USA, 2019. ACM.

[6] Likang Yin, Zhiyuan Zhang, Qi Xuan, and Vladimir Filkov. Apache software foundation incubator project sustainability dataset, 2021. Data provided by the Apache Software Foundation Incubator.