NAS journal
Hadhramout University

Contents lists available at https://digitalcommons.aaru.edu.jo/huj_nas/.

Hadhramout University Journal of Natural & Applied Science

Article

Digital Object Identifier:
Received 20 November 2024,
Accepted 5 December 2024,
Available online 25 January 2025

Data-Driven Strategies for Winning IoT Challenges: Case Study

Rasha A. Bin-Thalab[1,*], Taher A. Ghaleb[2]

[1]Department of Computer Engineering, Hadhramout University, Hadhramout, Yemen; r.binthalab@hu.edu.ye

[2]University of Trent, Peterborough, Canada; taherghaleb@trentu.ca

*Corresponding author: r.binthalab@hu.edu.ye

Abstract: In the competitive world of Internet of Things (IoT) endeavors, predicting a project’s success in contests can be challenging due to subjective and varied judging criteria. Our study addresses this problem by using machine learning to analyze outcomes in IoT contests, focusing on 104 competitions with a total of 5,863 projects from the Hackster.io platform. To the best of our knowledge, this is the first study to address this problem in the IoT community. We evaluated seven different machine learning models, which revealed that ensemble methods, such as random forest and gradient boosting, were the most effective, achieving an average accuracy of 80% and an Area Under the Curve (AUC) of 77%. We also performed a mixed-effects logistic regression that not only predicts a project’s likelihood of winning with an AUC of 86% but also uncovers the most significant factors that increase a project’s chances of success. These insights are valuable for IoT project creators, providing them with practical advice on how to improve their projects. Our research also offers useful insights for other stakeholders in the IoT community, such as IoT engineers and contest organizers, helping them understand what makes projects more likely to win. This study is a significant step in helping participants in IoT contests make informed decisions and increase their chances of success.

Keywords: Internet of Things; Online Contests; Hackster.io; Machine learning; Mixed-effects Logistic Regression

1. Introduction

IoT contests are dynamic platforms that foster innovation, creativity, and competition. Collaborating with sponsors, organizers present these challenges to attract a diverse group of innovators who strive to develop groundbreaking solutions. These contests not only address specific problems but also provide a stage for showcasing exceptional technical abilities. To encourage participation and innovation, organizers establish a framework with defined rules, deadlines, and rewards.

Hackster.io[(1)], a prominent platform in the IoT space, hosts dozens of these contests, attracting a wide range of participants, from expert engineers to ambitious beginners. The platform has become well-known for its reflection of the diverse range of skills, ideas, and innovation within the community. However, with such diversity comes the challenge of predicting project success in these contests. Due to the subjective and varied criteria employed by judging committees, determining the likely winners can be a challenging task for participants. This variability, compounded by the inherent subjectivity of human judgment, creates a level of unpredictability in these competitions.

Acknowledging the complexities in these IoT contests, this paper proposes a novel approach to analyze and forecast the results of these competitions using leveraging machine learning. Our approach makes a pivotal step towards understanding and predicting the dynamics of success in IoT contests, navigating through a landscape that has remained unexplored until now. Drawing from a rich dataset of 104 competitions and 5,863 projects on Hackster.io[(2)], we investigate the factors that contribute to a project’s success (win) or failure (lose). We apply seven different machine learning models to discover the superior performance of methods, and further employ mixed-effects logistic regression model to shed light on the key factors that enhance a project’s likelihood of winning.

This paper addresses two research questions (RQs) as follows:

RQ$_1$: How effectively can we predict the success of IoT projects in online contests?

The unpredictability of success in IoT contests, stemming from ambiguous evaluation criteria and the subjective nature of human judgment, poses a significant challenge. It is often difficult for IoT engineers to determine the winning potential of their projects or to understand past contest outcomes.

Therefore, in this RQ, we aim to predict whether an IoT project would win or lose an IoT contest. We employed seven machine learning models and evaluated them using five evaluation criteria with three different balancing techniques. This helps IoT engineers get an idea about the chances of their projects in a given contest before submission, thus making the waiting period less stressful.

RQ$_2$: What factors significantly influence success in online IoT contests?

Gaining insights into what drives success in IoT contests is crucial for IoT engineers looking to excel in these competitive platforms. However, assessing how projects align with judges' expectations remains challenging. Therefore, in this RQ, we investigate the most important features associated with contest winners. We employed a generalized logistic regression model to assess and identify the most important factors associated with IoT contest winning. This helps project owners focus more on the details that increase the chances for their projects to win a contest.

Overall, this research offers invaluable insights to various stakeholders in the IoT community. For project creators, it provides practical guidelines for enhancing their projects' chances of success. For IoT engineers and contest organizers, it offers a deeper understanding of the qualities that distinguish successful projects.

Overall, our research represents an important step towards equipping participants in IoT contests with the knowledge and tools to make more informed decisions and, ultimately, increase their chances of success in these competitive arenas.

The rest of this paper is organized as follows. Section 2 gives background on IoT, Hackster.io, and supervised machine learning. Section 3 describes how we collect and process the data used in our study. Section 4 presents our research questions, the approaches we used to address them, and the empirical findings for each. Section 5 discusses the implications of our findings. Section 6 presents the literature related to our study. Section 7 discusses the threats to the validity of our results. Finally, Section 8 concludes the paper and suggests possible future work.

2. Background

This section presents background on the Internet of Things and the Hackster.io community.

2.1 Internet of Things

The IoT is a burgeoning network where physical objects – devices, vehicles, buildings, and more – become interconnected. Embedded with sensors, software, and internet connectivity, these "things" bridge the gap between the physical and virtual worlds.

By enabling devices and machines to communicate and coordinate with each other, the IoT has the potential to revolutionize a wide range of industries, from smart homes and cities to industrial automation and healthcare. The seamless integration of physical and digital systems through the IoT can lead to increased efficiency, improved decision-making, and enhanced user experiences, ultimately making our lives more convenient and our world more connected [1]. Recent advancements have fueled the widespread adoption of IoT technologies. Increased availability, affordability, and scalability have made them more accessible. However, challenges (see [2] and [3]) remain as the diverse nature of IoT devices (heterogeneity) and the lack of established, widely adopted standards can complicate development and integration. To overcome these hurdles, collaboration among IoT engineers and practitioners is crucial. By sharing expertise, they can drive innovation and create more effective IoT solutions that benefit society.

2.2 Online IoT Communities

A vibrant ecosystem of online communities fosters knowledge sharing and collaboration in hardware and IoT development. Platforms like Hackster.io(3), Instructables(4), HackADay(5), and Hackr.io(6). connect both beginners and experienced professionals in the IoT field, allowing them to learn and grow from each other.

In this work, we use Hackster.io as our primary research source.

Hackster.io By 2024, Hackster.io has solidified its position as a leading online community for learning and sharing hardware and Internet of Things (IoT) development knowledge. This dynamic platform boasts a user base exceeding 1.6 million members, fostering a collaborative environment that empowers both budding hobbyists and seasoned professionals within the IoT field.

The Hackster.io community is remarkable in its depth and diversity, with over 20,000 active professional engineers contributing their knowledge and expertise. These experienced practitioners generously share their projects, ideas, and real-world experiences, fostering a robust learning ecosystem. Beyond just connecting people, the Hackster.io platform offers a wide range of content and features, including: an extensive project database, dedicated content channels covering news, events, and competitions related to the evolving IoT landscape

Hackster.io empowers IoT practitioners and enthusiasts to stay up-to-date, expand their skills, and collectively drive the progress of the IoT industry forward.

Online contests. There are various platforms for IoT project competitions, such as Hackster.io [4], Instructables [5], and Wiznet [6]. In these contest platforms, project owners of all ages from worldwide compete to demonstrate their skills and, most importantly, to attain reputation for their IoT projects. Similarly, there are other platforms for programming competitions, such as TechGig, that attract many competitive programmers of different ages and give them opportunities to train, learn, participate, and develop programming skills. Industrial Companies also find it very helpful to organize contests to overcome significant problems and introduce creative business solutions more quickly, avoiding traditional investments (e.g., R&D).

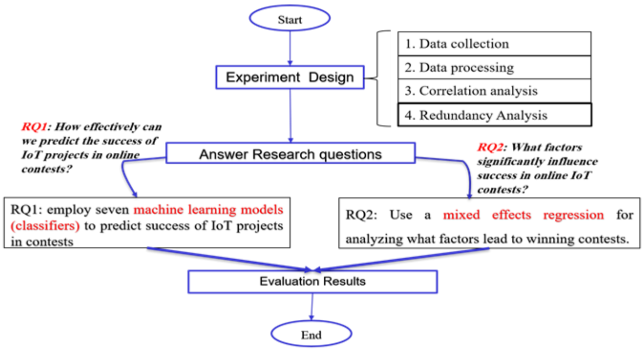

Start
Experiment Design
1. Data collection
2. Data processing
3. Correlation analysis
4. Redundancy Analysis
RQ1: How effectively can we predict the success of IoT projects in online contests?
Answer Research questions
RQ2: What factors significantly influence success in online IoT contests?
RQ1: employ seven machine learning models (classifiers) to predict success of IoT projects in contests
RQ2: Use a mixed effects regression for analyzing what factors lead to winning contests.
Evaluation Results
End

Figure 1. Flowchart of methodology steps

Hackster.io IoT contests. Hackster.io [4] hosts tens of IoT contests13 in which IoT engineers can submit their projects for chances to win prizes. Both projects and project owners can take advantage from participating in such contests to show of their IoT solutions to real world problems and build connections with the community. Each contest has an independent web page in which participants can find information about the rules of participation, number of submitted projects, deadlines, and also the final winners of the contests.

3. Methodology

We followed the analysis methodology to answer the research questions of this paper. Figure 1 summarizes steps followed by the researcher which includes: experiment design and evaluation results. The following subsections detail the steps involved.

3.1 Experiment Design

3.1.1 Data collection

We construct a crawler to collect data on all of IoT projects which participated in Hackster.io's contests [4]. Each project has several descriptive features such as a project link, a title, a description, tags, a connected channel, device elements, a story, the name of the competition in which it competed, the number of times it was submitted, and its competition rating, whether it won or not. To minimize the impact of the COVID-19 pandemic on project submissions, we focused our analysis on projects submitted between January 1, 2015 and December 8, 2020. This timeframe yielded data from 5,863 projects across 104 contests. We specifically chose Hackster.io due to its unique emphasis on learning and practical application of IoT concepts, making it distinct from generic online platforms. It goes beyond simply showcasing a vast array of hardware/software tools or training courses. Instead, the core focus is on well-documented, complete IoT projects designed to address specific challenges. This specialized approach allows Hackster.io to deliver a more curated and targeted experience for its users. Beyond project listings, the platform offers valuable metadata to empower informed decision-making.

3.1.2 Data processing

The first step is to perform an exploratory data analysis for a data science initiative (EDA). This involves learning more about the data that we are dealing with. We might define the shape of the data set (i.e., number of rows and columns), blank/null values, and visualize the correlation through the features to get other information. The data set contains 5,859 rows and 35 features. Table 1 lists the 35 features along with their data types (Categorical or Numeric) and descriptions. String and logical values are converted to Categorical values, whereas all numbers (integers or real numbers) are dealt with as Numeric. For example, the logical feature 'is winner' has two values:
'True' for winning a contest, or 'False' otherwise.

*1.3 Correlation Analysis***.**

To ensure the reliability of our regression models, we addressed the potential issue of multicollinearity, which occurs when independent variables are highly correlated. Following Harrell's guidelines [7], we performed a correlation analysis and employed Spearman rank ρ clustering analysis [8] using the varclus function from the rms R package. This analysis identified clusters of variables with strong correlations ($|\rho| > 0.7$).

Guided by the principle of parsimony (favoring simpler models) [9], we removed one variable from each such cluster while prioritizing those deemed more informative for predicting contest success. Our explanatory variables had similar complexity, so informativeness was the primary selection criterion. Figure 1 depicts the resulting dendrogram, highlighting the five clusters of highly correlated variables. Therefore, after removing highly correlated variables, we end up with 30 features.

*3.1.4 Redundancy Analysis***.**

Redundant variables can be estimated by other independent variables and, hence, may distort their relationships with the dependent variable (i.e., win or lose) [7]. Therefore, we analyze the independent variables used in our models to remove those that are redundant. we conducted a redundancy analysis to identify and remove any remaining redundant variables. We used the redun function from the rms R package [7]. This function assesses how well each remaining independent variable can be predicted by a combination of the others. If an independent variable has an R-squared (R^2) value greater than or equal to 0.9 (indicating it can be almost entirely explained by other variables), we excluded it from the model. This step ensures our final model includes only the most informative and independent variables for predicting contest success. Performing redundancy analysis revealed no redundant variables in our dataset.

4. Evaluation results

In this section, we discuss the motivation, approach, and findings of each of our research questions.

4.1 RQ1: How effectively can we predict the success of IoT projects in online contests?

4.1.1 Motivation

Due to unclear evaluation criteria and the subjectivity of human evaluation, it is hard to guess whether a project would win a contest. It is also challenging for IoT engineers to explore historical IoT contests to investigate why projects won or lost a contest. Hence, in this RQ, we aim to predict whether an IoT project would win or lose an IoT contest. This helps IoT engineers get an idea about the chances of their projects in a given contest before submission, thus making the waiting period less stressful.

4.1.2 Approach

To predict the success of IoT projects in contests, we compared seven different machine learning models (classifiers) using 30 calculated features (independent variables) extracted from the data. To prevent overfitting (models that perform well on training data but poorly on unseen data), we considered the Events Per Variable (EPV) ratio. In our context, EPV represents the number of projects per feature. A higher EPV indicates a lower risk of overfitting [10]. Reassuringly, our dataset has an EPV of 189, well above the recommended threshold of 10 [10].

We opted for Python as the programming environment due to its extensive libraries and frameworks for data manipulation and machine learning. Additionally, Python's large and supportive community, along with its comprehensive documentation, simplifies troubleshooting any challenges encountered during the experiments.

Selection of ML Models. We choose sever ML models, namely Random Forest (RF) [11], Gradient Boosting (XGB)[12], AdaBoost (Ada)[13], Decision Tree (DT)[14], Multilayer Perceptron (MLP) [15], Naive Bayes (NB)[16], and KNearest Neighbors (KNNs) [17]. The decision to employ a variety of models is driven by our aim to assess models with different natures and understand how their distinct mechanisms impact the prediction of success of IoT projects in online contests. Moreover, these models are among the most commonly used ML models in the literature for defect prediction problems. Specifically, our choice of RF and XGB is driven by their ensemble paradigm, which makes them robust against overfitting and ability to handle intricate data structures, but with distinct mechanisms: RF aggregates multiple decision trees to enhance accuracy and generalizability, while XGB iteratively corrects errors, boosting performance over time. Ada and DT also play crucial roles, as Ada enhances the performance of simple models, making it effective for diverse IoT projects, whereas DT is known for its interpretability, thus offering clear insights into how decisions are made based on the data features. We also employed MLP for its ability to model non-linear relationships, making it essential for our multifaceted dataset. Lastly, NB and KNNs were employed for their distinctive approaches: NB for its probabilistic and efficient nature in large datasets, and KNNs for their ability to make predictions based on local similarity, capturing subtle patterns that might be missed by more complex models. For all the models we employ, though we adhere to their default parameters, we specify some required parameters for the models to run. In particular, we set the split criterion using entropy for RF and DT, and the number of RF trees as 1, 000. We also set the number of estimators for XGB and Ada as 100. For MLP, we specify the size of hidden layers as 13 and the maximum number of iterations as 50. For KNN, we specify the number of neighbors as 7. The selection of the most suitable machine learning model is critical for accurate contest success prediction. Each model possesses distinct strengths and functionalities. By evaluating a diverse set of models (seven in this case), we ensure a thorough and comprehensive examination of the data. This multifaceted approach maximizes the likelihood of identifying the model that most accurately predicts winning IoT projects in online contests.

- Random Forest (RF): RF [11] is an ensemble learning method ideal for predicting the success of IoT projects in online contests. It constructs multiple decision trees during training, providing either class mode (classification) or mean prediction (regression) for input. RF’s capability to offer feature importance rankings is invaluable in analyzing the multifaceted nature of IoT projects, where numerous variables can influence success. Its ensemble approach reduces overfitting and improves generalization, crucial for capturing the diverse characteristics and complex patterns inherent in IoT project data. This model is particularly adept at handling the variety and intricacy of features that define successful IoT projects.
- Gradient Boosting (XGB): XGB [12], an advanced ensemble technique, builds trees sequentially to correct predecessors’ errors, enhancing accuracy iteratively. It is particularly efficient for the unbalanced nature of IoT contest data, where the number of winning and non-winning projects may vary significantly. Its focus on hardto-classify instances and error correction makes XGB ideal for our problem, where distinguishing subtle success factors in projects is key. The model’s interpretability and control over complexity, including regularization to prevent overfitting, are crucial for finely tuning predictions in the diverse world of IoT projects.
- AdaBoost (Ada): Another boosting method [13] that uses a set of weak learners (usually decision trees) to iteratively improve the predictions by assigning higher weights to the misclassified instances. AdaBoost improves the performance of simple models to boost overall accuracy, essential in the diverse environment of IoT contests. By adapting to the specific challenges of each project, AdaBoost can enhance prediction accuracy for various types of IoT projects, from simple gadgets to complex systems.
- Decision Tree (DT): DT [14] offers straightforward insights into decision-making processes based on data features. Its simplicity and interpretability are significant for dissecting the factors that contribute to a project’s success in IoT contests, providing clear, actionable insights into what makes a project more likely to win.
- Multilayer Perceptron (MLP): MLP [15], a type of neural network, that excels in identifying complex, non-linear relationships within datasets, a common scenario in IoT project success predictions. Its flexibility makes it an invaluable tool for uncovering the nuanced interplay of features that contribute to a project’s success in competitive environments.
- Naive Bayes (NB): NB [16] applies Bayes’ theorem with the assumption of conditional independence between features. Its simplicity and efficiency make it suitable for large datasets, typical in IoT project analysis. NB’s probabilistic approach is adept at handling the uncertainties and complexities inherent in predicting project success, where factors influencing outcomes can be numerous and interdependent.

- K-Nearest Neighbors (KNN): KNN [17] is a nonparametric algorithm that assigns an input to the class most common among its K-nearest neighbors based on feature similarity. In the context of IoT contests, KNN can effectively identify projects similar to past winners, leveraging patterns in historical data to predict future successes. This method’s reliance on the inherent data structure is particularly useful when projects exhibit distinct characteristics that correlate with winning outcomes Data balancing. The imbalance in our dataset, characterized by a significantly smaller number of winning projects compared to losing ones, poses a challenge for accurate model training. To address this, we implemented different data balancing techniques, each with its unique approach and implications:

- Imbalanced data: In this approach, we use the original dataset without any balancing modifications. This method maintains the dataset’s natural state, providing a baseline for evaluating the effectiveness of other balancing techniques. However, using imbalanced data can lead to biased models that favor the majority class, in this case, the losing projects, potentially reducing the predictive accuracy for the minority class, the winning projects.

- Oversampling (SMOTE): Synthetic Minority Over-sampling Technique (SMOTE) is an advanced oversampling method that creates synthetic samples for the minority class. It works by randomly choosing a point from the minority class and then creating new synthetic points along the line segments joining this point to its neighbors in the feature space. SMOTE helps balance the class distribution without losing valuable information, as it generates new, plausible examples of the minority class. This can improve model performance on the minority class but may introduce noise and risk of overfitting, as the model might become too tailored to the synthetic examples.

- Undersampling: This approach addresses imbalance by reducing the size of the majority class. It randomly eliminates samples from the majority class, thus equalizing the number of instances between classes. While under-sampling can effectively balance the dataset and reduce the training time, it also has the potential downside of losing potentially important information from the majority class. This loss of data can lead to increased variance in the model and may affect its ability to generalize well to new data.

Each of these techniques offers a distinct approach to handling the imbalance in our dataset, and their effectiveness can vary depending on the specific characteristics of the data and the models used. By comparing these techniques, we aim to identify the most effective method for improving the predictive performance of our models in the context of IoT project success prediction. Cross-validation for robust evaluation. To evaluate our ML models, we used various evaluation metrics to measure the performance of the models. To ensure our machine learning models were effectively trained and tested, we employed a widely recognized technique called 10-fold cross-validation. This method offers a balanced trade-off between bias and variance in model evaluation, leading to robust and reliable results. The process works as follows: the data is split into 10 statistically representative subsets (folds). In each iteration, the model is trained on 9 folds (90% of the data) and validated on the remaining fold (10%). This is repeated 10 times, ensuring each fold serves as the test set once. This maximizes data utilization while maintaining computational efficiency. Furthermore, we adopted a stratified k-fold approach. This ensures each fold maintains a similar proportion of projects that won and lost contests, mirroring the real-world distribution of successful and unsuccessful projects. This stratified approach guarantees a more thorough and representative evaluation of the model's performance across different data compositions [18].

By leveraging this well-established 10-fold cross-validation methodology, combined with a stratified data split, we were able to obtain robust and reliable results for our machine learning models. This rigorous evaluation process instills confidence in the models' ability to generalize and perform well on unseen data, making them suitable for real-world applications. Cross-validation offers several key advantages in our model evaluation process. Firstly, it exposes both the training and testing phases to diverse data patterns, mimicking real-world scenarios [10]. This helps prevent the model from overfitting to a specific subset of the data, ensuring a more robust and generalizable performance.

Moreover, the stratification of the cross-validation folds is crucial. This technique ensures that each fold reflects the overall composition of the dataset, maintaining the balance between successful and unsuccessful projects. By preserving the inherent distribution of the data, we can be confident that the model's performance is representative of its ability to handle the full spectrum of project outcomes. Another important aspect of our cross-validation approach is the use of a fixed random seed, in this case, 0. This allows for the reproducibility of our results, enabling other researchers to replicate our experiment and verify the findings. The ability to reproduce the results is a hallmark of reliable and rigorous research. Lastly, we calculated a comprehensive set of performance metrics to assess the effectiveness of our models during the cross-validation process. These metrics include accuracy, precision, recall, F1 score, and AUC-ROC. Each of these metrics provides a distinct perspective on the model's ability to accurately predict contest success. By considering multiple performance measures, we can gain a more nuanced understanding of the models' strengths and weaknesses. Accuracy [19] is a cornerstone metric in machine learning, gauging the proportion of projects (both winning and losing) the model correctly predicts. In our case, it reflects how well the model distinguishes between successful and unsuccessful IoT projects. A high accuracy score signifies the model's proficiency in differentiating winning from losing projects. This metric provides a general understanding of the model's overall prediction accuracy, regardless of the specific class (win or loss).

$$accuracy = \frac{Correct\ predictions}{All\ predictons} \qquad (1)$$

Table 1. Attributes description

	Feature	*Data Type*	*Description*
Contest Info	contest link	Categorical	URL link of a contest, used as a random effect in the mixed-effect logistic regression model
	num submissions	Numeric	The number of contest submissions (i.e., number of participated projects in each contest)
Project Info	difficulty level	Categorical	The difficulty level of a project: Beginner, Intermediate, Advanced, or Expert
	project type	Categorical	The type of a project: Work in progress, Protip, Showcase, Full instructions provided, Unknown
	Copyright	Categorical	The license of redistribution the project, e.g., Apache-2.0, CC BY-NC, GPL3+, etc.
	Likes	Numeric	The number of likes (i.e., likes a project received)
	project description length	Numeric	The number of words used to describe a project, which can go up to 142 Words
	Developers	Numeric	The number of developers working on a project, which can go up to ten
	estimated minutes	Numeric	the time (in minutes) required to build a project, which can go up to days
	Tags	Numeric	The number of tags per project, which can go up to 25 tags
	related channels	Numeric	The number of related channels per project, which can go up to 21 channels
	hardware quantity	Numeric	The number of all hardware quantity, which can go up to 635 items
	hardware items	Numeric	The number of unique hardware items used to build a project, which can go up to 61 items
	tool items	Numeric	The number of tool items per project, which can go up to 26 items
	software items	Numeric	The number of software items used per project, which can go up to 17 items
	purchase links	Numeric	The number of links to purchase hardware components or software apps and services per project, which can go up to 56
	vendors per item	Numeric	The number of vendors to purchase a single item per project, which can go up to 10 vendors per item
	unique hardware purchase sources	Numeric	The number of distinctive website links to purchase hardware components per project, which can go up to 24 links
	tools without links to purchase	Numeric	The number of tools that do not have a direct link to purchase, which can go up to 34
	story sections	Numeric	The number of sections in the Story part of a project, which explains how the project works, which can go up to 94, while some projects may have no story
	story length	Numeric	The length of the story, which can go up to 17K characters
	links in story	Numeric	The number of links in the story section, in this study there is up to 121 Links
	videos in story	Numeric	The number of videos attached to the project story, which can go up to 25
	images in story	Numeric	The number of images attached to the project story, which can go up to 171

	cad drawings in story	Numeric	The number of CAD drawings attached to the project story, which can go up to 29
Project Owner(s) Info	have personal websites	Categorical	Whether project owners have personal websites (True or False)
	project owners bio length	Numeric	The average length of a project owners' bio, which can go up to 29
	unique project owners projects	Numeric	The number of projects developed by project owners, which can go up to 60
	project owners followers	Numeric	The number of distinct followers of project owners, which can go up to 1K
	project owners tools	Numeric	The number of distinct tools utilized by project owners should be indicated, which can go up to 338 tools
	project owners channels	Numeric	The number of distinct channels project owners may have, which can go up to 425
	project owners awards	Numeric	The number of distinct awards received by project owners, which can go up to 27 contests
	project owners comments	Numeric	The number of comments project owners received, which can go up to 483 Comments
	project owners given likes	Numeric	The number of likes project owners gave to other projects or posts, which can go more than 2K likes
	project owners received likes	Numeric	The number of likes the project owners received from other users, which can go up to 183 likes

• Precision [19]: It focuses on the predicted "winning" projects and calculates the proportion of true positives – TP (correctly predicted winning projects) to the sum of true positives and false positives – FP (losing projects incorrectly predicted as winning). Precision reflects the reliability of the model when it predicts a project as a winner. High precision implies that a project predicted to win is likely to be genuinely successful.

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

• Recall [19]: It measures the proportion of true positive predictions – TP (correctly predicted winning projects) to the sum of true positives and false negatives – FN (winning projects incorrectly predicted as losing). Recall indicates the model's ability to capture and correctly predict all actual winning projects. A high recall means that the model effectively identifies most winning projects, ensuring minimal missed opportunities.

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

• F1-score [19]: It is the harmonic mean of precision and recall, providing a balance between the two. In scenarios with uneven class distribution (more losing than winning projects or vice versa), achieving a balance between precision and recall is crucial. The F1 score encapsulates this balance, ensuring that both false alarms (wrongly flagged winning projects) and missed winning projects are minimized.

$$F1\_score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \qquad (4)$$

•

• The AUC-ROC [19] (Area Under the ROC Curve) metric goes beyond simple accuracy. It assesses the model's ability to rank winning projects (positive instances) higher than losing projects (negative instances). Imagine randomly picking two projects: one a winner and one a loser. A high AUC score indicates the model is more likely to rank the winner higher, regardless of a specific threshold used to classify success. This is crucial in IoT contests, where the boundary between success and failure can be blurry. A strong AUC score signifies the model's ability to make reliable predictions that hold true even if the exact threshold for "winning" is adjusted. This flexibility is valuable in real-world scenarios where the optimal threshold may depend on specific needs.

4.1.3 Findings

Figures 2, 3, and 4 show the performance results of the seven models we employ for predicting contest winning of IoT projects using the 30 features and the three balancing techniques: imbalanced (Figure 3), SMOTE (Figure 4), and undersampling (Figure. 5).

Finding 1.1: Best performing model. The results show that RF and XGB are the best-performing models in terms of accuracy, F1-score and AUC, regardless of the balance type. They achieve an average accuracy of 80%, an average F1-score of 65% and an average AUC of 77%. Ada and DT are the next best models, with an average accuracy of 78%, an average F1-score of 62% and an average AUC of 75%.

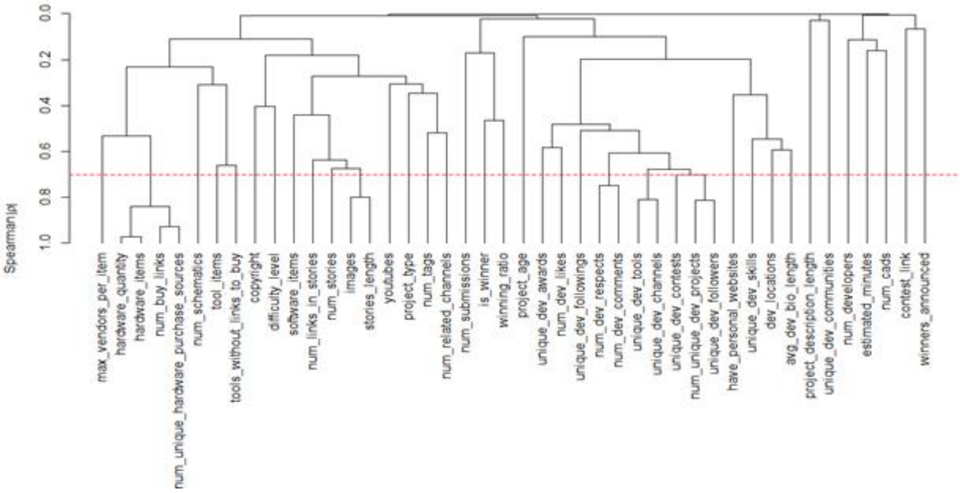


Figure 2. The hierarchical clustering, correlation pairs, of features

MLP and NB achieved the worst performance, with an average accuracy of 59%, an average F1-score of 38% and an average AUC of 56%. K-Nearest Neighbors is the mid-performing model, with an average accuracy of 63%, an average F1-score of 34% and an average AUC of 55%. The results suggest that ML models can be used to predict the success of IoT projects in online contests based on various perspectives, such as technical quality, novelty, social impact, etc. This can help IoT engineers improve their project design and development, IoT project owners optimize their project presentation and promotion, contest organizers evaluate and rank project submissions, and other stakeholders identify and support promising IoT projects.

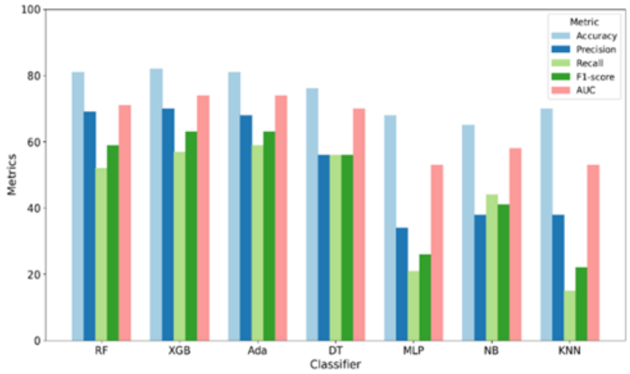


Figure 3. Evaluation results using imbalanced data for the seven.

However, the results also indicate that there is room for improvement in the prediction performance of the ML models, especially in terms of precision and recall. Therefore, further research is needed to explore more features, more models, and more ways to collect balanced data for predicting the contest winning of IoT projects.

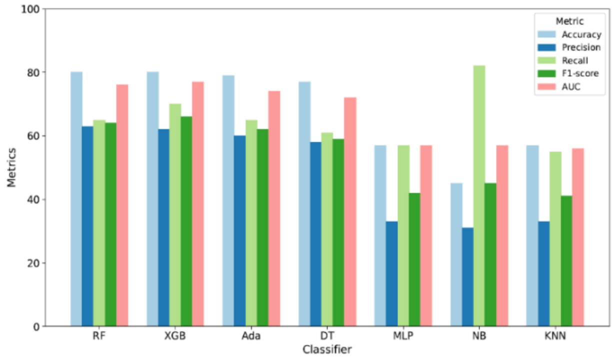


Figure 2. Evaluation results using SMOTE-balanced data for the seven ML models.

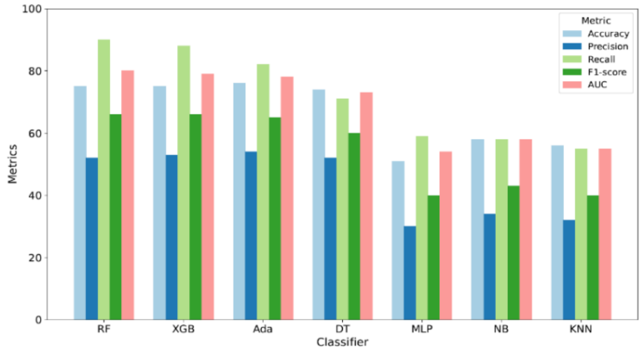


Figure 5. Evaluation results using Undersampling-balanced data.

Finding 1.2: Impact of data balancing**.** Results show that balancing techniques has a significant impact on the performance of the models. In particular, SMOTE and undersampling improve the recall and AUC of the models, but decrease their precision and accuracy. This is because they increase the number of true positives, but also increase the number of false positives, making them better at predicting winning projects than losing projects, but they also make more mistakes in both classes. SMOTE achieved slightly better performance, making it more effective as it preserves more information from the original data and generates more realistic synthetic samples. On the other hand, when no balancing technique is applied, models tend to have higher accuracy and precision but lower recall and AUC, making them more applicable at predicting losing projects than winning projects, which is expected given that the data is imbalanced and there are more losing projects than winning projects. Therefore, in practice, the optimal balance type would depend on the tradeoff between precision and recall that is desired for the prediction task.

Finding 1.3: Performance trade-offs in our models. The trade-offs between precision and recall present crucial considerations for the deployment of our models in practice. Our results show that, despite its comparatively lower accuracy, NB achieves a high precision of 82%, indicating its strong ability to correctly predict winners. However, this comes with a lower recall of 45%, suggesting that it might miss many potential winners while also incorrectly identifying unsuccessful projects as winners. Conversely, XGB excels in recall with a remarkable rate of 70%, demonstrating its effectiveness in capturing a large portion of actual winning projects. This higher recall, paired with a precision of 62% and an F1-score of 66%, indicates a well-balanced approach, making XGB a reliable choice in scenarios where identifying as many true winners as possible is crucial. RF presents a slightly different balance, with a recall of 65% and a precision of 63%, accompanied by an F1-score of 64%. This suggests that RF is slightly more conservative than XGB in predicting winners, making it suitable for situations where a slightly higher precision is preferred, albeit at a small cost to recall. In practical terms, the choice between XGB and RF would depend on the specific priorities of the IoT contest. If maximizing the identification of winning projects is important, XGB would be ideal given its higher recall and balanced F1-score. In contrast, for contexts where reducing false positives is slightly more important, RF's balance of precision and recall would be more appropriate. These models offer distinct advantages under varying conditions, underscoring the importance of aligning model selection with the strategic objectives and constraints of IoT contest predictions.

Finding 1.4: Performance trade-offs in our models. Understanding model uncertainties is crucial for assessing their applicability in practice. To this end, we investigate the uncertainties of our models by analyzing the degree of errors or incorrect predictions represented by false positives and false negatives. Our results show that both RF and XGB exhibit a balance of relatively lower false positive and false negative rates. Specifically, XGB has a false positive rate of 16%, while Random Forest RF has a slightly lower v of 14%. This indicates that XGB, though excellent in identifying winning projects, might also suffer from misclassifying losing projects as winners. In practical scenarios, a higher false positive rate might be acceptable in contexts where the emphasis is on not smissing any potential winners, even if it means including some less promising projects. This approach could be beneficial in early stages of contests where casting a wider selection is more important than precision. On the other hand, XGB has a false negative rate of 30%, which is lower than RF's 35%, indicating that XGB is more adept at capturing true winners. A lower false negative rate is crucial in scenarios where missing out on a genuine winning project carries a significant opportunity cost. In high-stakes contests or when filtering projects for limited resources or attention, a model like XGB would be preferable due to its lower likelihood of overlooking winning entries. The choice of model, therefore, should be tailored to the specific priorities and resource considerations of the IoT contest environment, balancing the need to discover potential winners against the practicalities of resource allocation and project support.

4.2 RQ$_2$: What factors significantly influence success in online IoT contests?

4.2.1 Motivation

IoT engineers are in need of information that makes their projects successful, especially when it comes to participating in contests. However, it is challenging for IoT engineers to assess how their projects would be perceived and evaluated by judges. Therefore, in this RQ, we investigate the most important features associated with contest winners. This helps project owners focus more on the key details that increase the chances for their projects to win a contest.

4.2.2 Approach

Our dataset encompasses 5,859 projects across 104 contests. Since each contest has unique schedules and selection criteria, we use a mixed-effects regression approach to account for these variations in winning probabilities between contests.

To understand how the features listed in Table 1 influence contest success, we specifically employ generalized mixed-effects models for logistic regression. These models combine fixed and random effects [20]:

Fixed effects: Represent variables with consistent effects across all projects (e.g., a specific feature's impact).

Random effects: Account for variations between groups (contests) in this case, capturing the differing winning probabilities across contests.

Our models incorporate a random intercept for each project [21]. This allows us to consider the inherent differences in winning chances between contests, providing a more nuanced picture.

Traditional models (like those used in RQ1) only use fixed effects, neglecting these contest-to-contest variations.

For a deeper understanding, the mathematical formula for the mixed-effects logistic model is provided in Equation 5. Here, Y_g represents the win/lose outcome, β_0 is the constant intercept, X_i are the independent variables, β_i are their coefficients, ϵ_g represents errors, and θ_g represents the varying intercepts for each project.

We implemented these models using the glmer function from the lme4 R package, specifying the appropriate parameters.

$$Y_g = \beta_0 + \theta_g + \sum_{i=1}^{n} \beta_i X_i + \epsilon_g \qquad (5) \ [22]$$

By employing mixed-effects regression, we effectively account for contest variations, enabling a more robust and insightful exploration of how features influence project success

Mixed-effects logistic regression models use asterisks (*) to highlight statistically significant independent variables. This significance is determined based on an ANOVA test [23]. An independent variable is considered significant if its p-value (written as $\Pr(>|\chi^2|)$) is less than 0.05. The p-value represents the probability of observing a chi-squared (χ^2) statistic as extreme as the one calculated, assuming the variable has no effect. In simpler terms, a low p-value (less than 0.05) suggests the variable is unlikely to be random and likely has a true influence on the outcome.

The model also uses arrows (↗ upward or ↘ downward) to indicate the direction of the relationship between the independent variable and the outcome (winning the contest). An upward arrow (↗) signifies a positive or direct relationship, meaning higher values of the variable are associated with a greater likelihood of winning. Conversely, a downward arrow (↘) indicates an inverse relationship, where higher values of the variable lead to a lower chance of winning.

To assess the predictive performance of the mixed-effects logistic regression models in forecasting contest winners, we employ three key evaluation metrics:

- Area Under the Curve (AUC): The AUC metric provides a measure of the model's ability to discriminate between contest winners and non-winners. A higher AUC value indicates better predictive power [24].
- Marginal R^2: The marginal R^2 represents the proportion of variance in the outcome that is explained by the fixed effects (i.e., the independent variables) in the model. This metric quantifies the overall explanatory power of the model's predictors [25].
- Conditional R^2: The conditional R^2 takes into account both the fixed effects and the random effects (i.e., the grouping variables) in the model. It represents the proportion of variance in the outcome that is explained by the entire mixed-effects model, including both the fixed and random components [20].

These three performance measures - AUC, marginal R^2, and conditional R^2 - allow us to comprehensively evaluate the effectiveness of the mixed-effects logistic regression models in predicting the outcomes of the contests.

The mixed-effects logistic regression model estimates coefficients for each independent variable. These coefficients, either positive or negative, reveal the direction of the relationship between the variable and the likelihood of winning the contest.

- Positive Coefficient (↗): This indicates a direct relationship. As the value of the variable increases, the odds of winning also go up.
- Negative Coefficient (↘): This signifies an inverse relationship. A higher value of the variable is associated with a lower chance of winning.

While coefficients tell us the direction of the effect, odds ratios [26] provide a more precise measure. These ratios quantify the association between a variable and winning, holding all other variables constant. For instance, an odds ratio of 2 for a variable suggests that the odds of winning are multiplied by 2 for every unit increase in that variable.

In essence, coefficients show the direction (positive or negative), and odds ratios quantify the magnitude of the effect on the likelihood of winning the contest. This combined approach provides a clearer understanding of how each variable influences the outcome.

4.2.3 Findings:

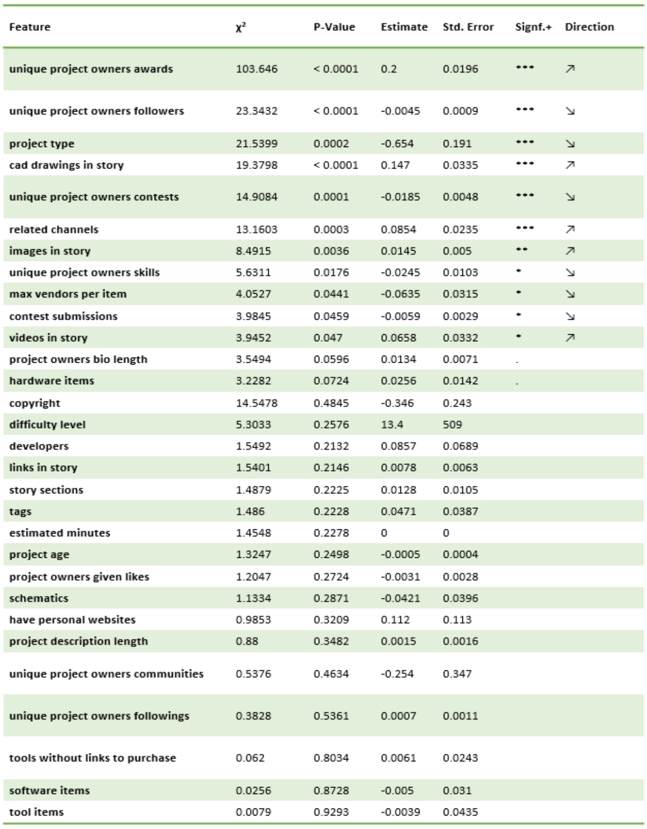
Finding 2.1: Our models achieve an AUC of 86% in predicting contest winning IoT projects. The overall AUC of 86% reveals the high predictive power achieved by our model. In addition, we find the conditional R2 of 50%, which is as twice as the marginal R2 of 25%. This result indicates that our model is sensitive to the variances in contests, i.e., contest winning could be different from one contest to another, depending on many factors, such as judgement criteria.

Finding 2.2: The more awards and contest participation an IoT engineer has, the more chances to win new IoT contests. Our analysis of the important features of modelling contest winning shows that IoT engineers who are more familiar with participating in online contests tend to increase the likelihood of winning future contests. In particular, our model reveals that being an award-winning IoT engineer is the top most significant factor associated with making an IoT project wins a contest (a $\chi 2$ of 103.646, which represents 39% of the overall model predictive power). This encourages IoT engineers that collaborate with those who have more experience in this context to make their projects more competitive. However, we observe that having more followers on Hackster.io has, surprisingly, an inverse association with contest winning. This suggests that projects submitted by popular IoT engineers are unlikely to win contests. Our investigation of example projects shows that some projects are submitted by IoT industries, such as Arduino, making them in some way ineligible to participate in contests. Therefore, beginning IoT engineers are recommended to participate in IoT contests regardless of who else is participating. IoT engineers should pay more attention to the details of their projects than to who is participating in a contest.

Finding 2.3: IoT projects with more enriched page content are highly likely to win IoT contests. We observe that projects in which project Story have more videos, images, and/or CAD drawings have a higher likelihood of winning contests. This indicates the importance of providing details on various types of information, which helps other IoT engineers understand and reproduce IoT projects, and also makes projects more appreciable by the community, including contest judges.

Finding 2.4: The number of submission to IoT contests has a significantly negative relationship with contest winning. When many IoT projects over participate in IoT contests, their chances of winning are likely to decrease, as multiple submissions might indicate that a project (a) participates in any contest regardless of relevance, (b) has failed in other contests, or (c) has already won other contests. As a result, contest judges may notice such behavior, and hence prefer to give a chance to other first-time participating projects. Therefore, project owners should only participate in very relevant contests to make room for other potential projects.

Finding 2.5: Protip and in-progress projects have lower chances of winning IoT contests. Compared to projects with Full instructions, protip and in-progress projects have a lower chance of winning IoT contests. The percentages range from 17 and 19 for protip and in-progress, while it rises to 29 when complete instructions are provided, as shown in Table 3. Therefore, project owners are encouraged to describe their projects more comprehensively to attract other community users as well as contest judges.



Feature	χ^2	P-Value	Estimate	Std. Error	Signf.+	Direction
unique project owners awards	103.646	< 0.0001	0.2	0.0196	***	↗
unique project owners followers	23.3432	< 0.0001	-0.0045	0.0009	***	↘
project type	21.5399	0.0002	-0.654	0.191	***	↘
cad drawings in story	19.3798	< 0.0001	0.147	0.0335	***	↗
unique project owners contests	14.9084	0.0001	-0.0185	0.0048	***	↘
related channels	13.1603	0.0003	0.0854	0.0235	***	↗
images in story	8.4915	0.0036	0.0145	0.005	**	↗
unique project owners skills	5.6311	0.0176	-0.0245	0.0103	*	↘
max vendors per item	4.0527	0.0441	-0.0635	0.0315	*	↘
contest submissions	3.9845	0.0459	-0.0059	0.0029	*	↘
videos in story	3.9452	0.047	0.0658	0.0332	*	↗
project owners bio length	3.5494	0.0596	0.0134	0.0071	.	
hardware items	3.2282	0.0724	0.0256	0.0142	.	
copyright	14.5478	0.4845	-0.346	0.243		
difficulty level	5.3033	0.2576	13.4	509		
developers	1.5492	0.2132	0.0857	0.0689		
links in story	1.5401	0.2146	0.0078	0.0063		
story sections	1.4879	0.2225	0.0128	0.0105		
tags	1.486	0.2228	0.0471	0.0387		
estimated minutes	1.4548	0.2278	0	0		
project age	1.3247	0.2498	-0.0005	0.0004		
project owners given likes	1.2047	0.2724	-0.0031	0.0028		
schematics	1.1334	0.2871	-0.0421	0.0396		
have personal websites	0.9853	0.3209	0.112	0.113		
project description length	0.88	0.3482	0.0015	0.0016		
unique project owners communities	0.5376	0.4634	-0.254	0.347		
unique project owners followings	0.3828	0.5361	0.0007	0.0011		
tools without links to purchase	0.062	0.8034	0.0061	0.0243		
software items	0.0256	0.8728	-0.005	0.031		
tool items	0.0079	0.9293	-0.0039	0.0435		

Table 3. Project winning percentages by type

Project type	Winning percentage
Protip	19
Work in progress	17
Full instructions provided	29

5. Discussion This section discusses the implications of our research for IoT engineers and contest organizers. Likelihood of contest winning. Our model achieved a high AUC value of 86%, which gives IoT engineers early feedback on whether to participate in an online contest or not. If a project is likely to win, IoT engineers should get more confidence to participate their project at that online contest. Otherwise, IoT engineers should work hard on improving their projects to satisfy the least required factors of winning to compete with other participating projects. This not only helps for contest participation, but also the overall quality of IoT projects hosted on online communities.

Winning factors. Winning or losing IoT contests sometimes happens due to luck, connections, bias, or subjectivity. Yet, there are always strong points that can make projects successful no matter what. Instead of random guessing whether a project would win a contest or not, our model reveals the most important factors associated with contest winning. IoT engineers have control over most of these factors, such as enriched content, full instructions, etc. Yet, satisfying these factors does not always guarantee contest winning, given the limited number of winners in each contest. Therefore, IoT engineers should pay attention to other contest-related factors, such as deadline, number of submissions, and contest relevance, before moving forward with participation. Moreover, even if a project loses at one contest, it may win at another contest. However, IoT engineers should be very careful about submitting to multiple contests, as losing many times can be a bad indication about that project, as our results indicate. Transparent contest judgement. Participation criteria in online contests are usually written in a traditional way that makes it difficult for IoT engineers to know what exactly they need to work on to get better chances of winning a contest. Contest organizers should make the judgement process more transparent by whether sharing detailed scores about what helped and what did not help a project win or lose a contest. This can help IoT engineers learn from their previous mistakes and improve in the future. It also suggests that IoT contest organizers should share such data, anonymously, with any future contests to encourage other online contests to adopt such transparency in their judgement.

6. Related Work

In this section, we present the existing work related to (i) IoT technology and (ii) prediction of contest winners.

6.1 IoT studies

The Internet of Things (IoT) has captured the attention of researchers across various domains [27]. Studies have explored a diverse range of topics, including context-aware approaches, fault tolerance in IoT services, the integration of IoT and cloud computing, IoT service composition, and the popularity of different IoT projects.

- Context-aware IoT approaches aim to develop systems that can adapt to their surroundings. For example, Chattopadhyay et al. [28] proposed a method to simplify building context-aware IoT applications without specialized knowledge. D'Oca et al. [29] developed a framework using data mining to analyze window opening/closing behavior based on factors like temperature and occupancy, demonstrating the potential of context-aware IoT to enhance user experience and optimize resource use.
- Fault Tolerance: Su et al. [30] propose a method enabling rapid failover mechanisms (recovery within seconds) upon replacing faulty IoT devices, eliminating the need for manual intervention. This significantly improves system uptime and resiliency.
- IoT & Cloud Integration: Botta et al. [3] identify a need for greater standardization within both IoT and cloud computing to facilitate seamless integration. This would enable smoother data exchange and processing between these domains.

- IoT Service Composition: Tzortzis et al. [31] propose a semi-automatic approach to assist project owners in discovering, utilizing, and connecting various IoT services for building more complex functionalities. This can streamline the development of sophisticated IoT applications.
- Social Media & IoT: Ustek-Spilda et al. [32] analyze active social media discussions on IoT in Europe. Their findings suggest users within the same geographical region are more likely to connect online regarding IoT, and hashtags related to IoT technology show high correlation. This indicates potential for geographically targeted marketing and community building within the IoT space.
- IoT in Healthcare: The medical field is a leading adopter of IoT technology [33]. Gómez et al. [34] propose an ontology-based architecture for managing fitness and exercise programs, aiming to provide personalized guidance for patients with chronic illnesses. Additionally, Ghaleb et al. [35] analyze popular IoT projects on Hackster.io to identify characteristics that contribute to project success, offering valuable insights for developers in the field. Ghaleb et al. [36] also studied online IoT communities to characterize IoT projects that were developed in response to the COVID-19 pandemic. Recent studies, such as Clemente-Lopez's work [37], have focused on implementing security measures in IoT healthcare systems. One notable approach involves utilizing chaos-based encryption to safeguard sensitive data transmitted by wearable devices.

Unlike the aforementioned studies, our study focuses on the features that distinguish an IoT project from other projects in terms of likelihood of winning contests.

6.2 Prediction of contest winners

In many applications, such as hackathons, sports, games, and even policy, predicting a winner has become an important research topic. Such predictions are usually performed using different metrics, such as Confidence-Calibrated [38], Sentiment analysis [39], machine learning techniques. Ravari et al. [40] proposed a model to predict a winner in a game, and provided detailed analysis of the features that help predict winners in the StarCraft game. The authors split features into time-dependent and time-independent groups and measured the mean, the variance, and the difference between the two players. In order to calculate the analysis of the relative importance of the features and prediction accuracy, the Random Forest and Gradient Boost classification was implemented. The results showed about 63% for accuracy, and economic feature got the highest importance among other time dependent features to predict the winner in the StarCraft game. Another study by Demchuk [41] on hackathons. The authors implemented the machine learning strategies to illustrate the importance of various project features by providing analysis of large hackathon dataset. They used Naïve Bayes, Logistic Regression, and Random Forest for prediction of wining hackathon projects. Furthermore, for a better prediction, they concentrate on project feature extraction and feature selection. In another field, Lee with his colleague [42] conduct a study to examine the relationship between learning performance—encompassing acquisition and reversal learning—in domestic pigs and their success in competitions against unfamiliar opponents. Next, another work [43] investigates the impact of penalties and scores on the outcome of elite judo contests, exploring how these variables predict the final result.

Unlike the aforementioned studies, our study focuses on predicting winning projects in IoT contests using features that correspond to projects themselves as well as contests.

7. Threats to Validity

This section discusses the potential threats to the validity of our work.

7.1 Internal Validity

Internal threats to validity are concerned with the ability to draw conclusions from the attributes of the projects in our dataset [44]. We computed the factors at the project-level, including contest-related factors. This can make contest-related factors repeated across projects, which could affect their importance in our model. We mitigate this issue by controlling our results by contest by making a contest as a random effect in our mixed-effects logistic regression model. In addition, our results are based on the 35 features we computed, which might not be comprehensive enough to capture all project characteristics. We aim in the future expand our feature scope to consider additional contest and project-related features by getting feedback from contest organizers in this contest.

7.2 External Validity

This section addresses the potential limitations of generalizing our research results (external validity) [44]. Our conclusions are based on data collected from 5,863 IoT projects on Hackster.io, which participated in 104 IoT contests within the interval 2015 to 2020. However, these findings might not be directly applicable to other online communities, other datasets, or other time periods.

To address the generalizability challenge, we attempted to explore other platforms such as Instructables and HackADay. Unfortunately, these websites encompass a broader range of content beyond just IoT projects, including hardware, software, and training courses. Additionally, unlike Hackster.io, these platforms lack features that specifically distinguish high-quality IoT projects from less successful ones, such as estimated project reproduction time, number of "respects" received by a project (a measure of community recognition), project difficulty level, and comprehensiveness of project instructions provided by the owner.

The significant variation in website design and structure across platforms would necessitate the development of custom data crawlers for each one, making a broader analysis a substantial undertaking. As a result, we acknowledge that our findings may not be fully generalizable to all IoT communities and platforms, and further research may be needed to explore the broader applicability of our conclusions.

8. Conclusion

In our comprehensive empirical study, we harness the capabilities of machine learning to address the complex dynamics of winning in IoT contests within online communities, a topic that has remained largely unexplored in the IoT domain. By analyzing historical data from 104 contests, encompassing a diverse array of 5,863 IoT projects on the Hackster.io platform, we provide valuable insights into the factors that influence project success in these competitive environments. Our investigation, employing seven distinct machine learning models, revealed that ensemble methods, particularly Random Forest (RF) and Gradient Boosting (XGB), stand out with their superior performance, achieving a high average prediction accuracy of 80% and an Area Under the Curve (AUC) of 77%. These findings highlight the effectiveness of these methods in navigating the complex and subjective criteria that define IoT contest success. Furthermore, the development of a mixed-effects logistic regression model marked a significant advancement in our study. This model not only efficiently predicted the likelihood of a project's success with an AUC of 86% but also unveiled the most important factors that significantly boost a project's chances of winning. These insights are invaluable for IoT project creators, equipping them with practical strategies to enhance their projects' potential for success. Our research has important implications for the broader IoT community, including project creators, engineers, and contest organizers, by shedding light on the attributes that contribute to a winning project. This enhanced understanding is vital for participants seeking to make informed decisions and improve their odds of success in IoT contests.

In the future, we plan to engage with contest organizers to gather feedback on our findings, further enriching the depth and applicability of our research. Additionally, we intend to delve into the perspectives of contest judges to gain a more nuanced understanding of their decision-making criteria. Expanding our evaluation to encompass other online IoT communities and fine-tuning large language models (LLMs) to recognize patterns of successful projects will be key areas of our future research endeavors. Our ultimate goal is to refine and adapt these models to better predict and understand the success factors in the ever-evolving landscape of IoT competitions.

Margins

(1) https://www.hackster.io
(2) https://www.hackster.io/contests
(3) https://www.hackster.io
(4) https://www.instructables.com
(5) https://hackaday.com
(6) https://Hackr.io

References:

[1] F. Mattern and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," in *From active data management to event-based systems and more*, Springer, 2010, pp. 242–259.

[2] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, 2015.

[3] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "On the integration of cloud computing and internet of things," in *International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2014, pp. 23–30.

[4] "Hackster.io - The community dedicated to learning hardware." Accessed: Jun. 14, 2024. [Online]. Available: https://www.hackster.io/

[5] "Yours for the making - Instructables." Accessed: Jun. 14, 2024. [Online]. Available: https://www.instructables.com/

[6] "WIZnet Official - Leading Internet Connectivity Solutions." Accessed: Jun. 14, 2024. [Online]. Available: https://wiznet.io/

[7] F. E. Harrell, "Regression modeling strategies, with applications to linear models, survival analysis and logistic regression," *GET ADDRESS Springer*, 2001.

[8] W. Sarle, "The VARCLUS Procedure.," *SASSTAT Users Guide*, 1990.

[9] J. Vandekerckhove, D. Matzke, and E.-J. Wagenmakers, "Model Comparison and the Principle," in *The Oxford handbook of computational and mathematical psychology*, vol. 300, Oxford Library of Psychology, 2015.

[10] P. Peduzzi, J. Concato, E. Kemper, T. R. Holford, and A. R. Feinstein, "A simulation study of the number of events per variable in logistic regression analysis," *J. Clin. Epidemiol.*, vol. 49, no. 12, pp. 1373–1379, 1996.

[11] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," *Int. J. Data Sci. Anal.*, pp. 1–15, 2024.

[12] N. Gunasekara, B. Pfahringer, H. Gomes, and A. Bifet, "Gradient boosted trees for evolving data streams," *Mach. Learn.*, vol. 113, no. 5, pp. 3325–3352, 2024.

[13] R. E. Schapire, "Explaining adaboost," in *Empirical inference: festschrift in honor of vladimir N. Vapnik*, Springer, 2013, pp. 37–52.

[14] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[15] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, no. 7, pp. 579–588, 2009.

[16] O. Peretz, M. Koren, and O. Koren, "Naive Bayes classifier–An ensemble procedure for recall and precision enrichment," *Eng. Appl. Artif. Intell.*, vol. 136, p. 108972, 2024.

[17] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers-a tutorial," *ACM Comput. Surv. CSUR*, vol. 54, no. 6, pp. 1–25, 2021.

[18] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, Montreal, Canada, 1995, pp. 1137–1145.

[19] R. Graf, M. Zeldovich, and S. Friedrich, "Comparing linear discriminant analysis and supervised learning algorithms for binary classification—A method comparison study," *Biom. J.*, vol. 66, no. 1, p. 2200098, 2024.

[20] J. J. Faraway, *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*, vol. 124. CRC press, 2016.

[21] B. Winter, "A very basic tutorial for performing linear mixed effects analyses," *ArXiv Prepr. ArXiv13085499*, 2013.

[22] D. Bates *et al.*, "Package 'lme4,'" *convergence*, vol. 12, no. 1, p. 2, 2015.

[23] P. Pinheiro, "Linear and nonlinear mixed effects models. R package version 3.1-97," *Httpcran R-Proj. Orgwebpackagesnlme*, 2010.

[24] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[25] S. Nakagawa and H. Schielzeth, "A general and simple method for obtaining R2 from generalized linear mixed-effects models," *Methods Ecol. Evol.*, vol. 4, no. 2, pp. 133–142, 2013.

[26] A. Agresti, "Tutorial on modeling ordered categorical response data.," *Psychol. Bull.*, vol. 105, no. 2, p. 290, 1989.

[27] M. Kumar, P. K. Singh, M. K. Maurya, and A. Shivhare, "A survey on event detection approaches for sensor based IoT," *Internet Things*, vol. 22, p. 100720, 2023.

[28] T. Chattopadhyay, S. Banerjee, S. Maiti, S. Dey, D. Jaiswal, and B. Barik, “Way to make ourselves redundant: A Semantic Framework for Automated Workflow Generation for IoT,” *TCS Tech. Archit.*, 2015.

[29] S. D’Oca and T. Hong, “A data-mining approach to discover patterns of window opening and closing behavior in offices,” *Build. Environ.*, vol. 82, pp. 726–739, 2014.

[30] P. H. Su, C.-S. Shih, J. Y.-J. Hsu, K.-J. Lin, and Y.-C. Wang, “Decentralized fault tolerance mechanism for intelligent IoT/M2M middleware,” in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, IEEE, 2014, pp. 45–50.

[31] G. Tzortzis and E. Spyrou, “A semi-automatic approach for semantic IoT service composition,” in *Workshop on Artificial Intelligence and Internet of Things in conjunction with SETN*, 2016.

[32] F. Ustek-Spilda *et al.*, “A twitter-based study of the European internet of things,” *Inf. Syst. Front.*, pp. 1–15, 2020.

[33] W. A. Al-Nbhany, A. T. Zahary, and A. A. Al-Shargabi, “Blockchain-IoT healthcare applications and trends: a review,” *IEEE Access*, 2024.

[34] J. Gómez, B. Oviedo, and E. Zhuma, “Patient monitoring system based on internet of things,” *Procedia Comput. Sci.*, vol. 83, pp. 90–97, 2016.

[35] T. A. Ghaleb, D. A. da Costa, and Y. Zou, “On the Popularity of Internet of Things Projects in Online Communities,” *Inf. Syst. Front.*, 2021, doi: 10.1007/s10796-021-10157-1.

[36] T. A. Ghaleb, R. A. Bin-Thalab, and G. A. A. Alselwi, “How Internet of Things responds to the COVID-19 pandemic,” *PeerJ Comput. Sci.*, vol. 7, p. e776, 2021.

[37] D. Clemente-Lopez, J. de Jesus Rangel-Magdaleno, and J. M. Muñoz-Pacheco, “A lightweight chaos-based encryption scheme for IoT healthcare systems,” *Internet Things*, vol. 25, p. 101032, 2024.

[38] D.-H. Kim, C. Lee, and K.-S. Chung, “A confidence-calibrated moba game winner predictor,” in *2020 IEEE Conference on Games (CoG)*, IEEE, 2020, pp. 622–625.

[39] L. Oikonomou and C. Tjortjis, “A method for predicting the winner of the usa presidential elections using data extracted from twitter,” in *2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_CECNSM)*, IEEE, 2018, pp. 1–8.

[40] Y. N. Ravari, S. Bakkes, and P. Spronck, “Starcraft winner prediction,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2016.

[41] S. Demchuk, “Predicting hackathon outcomes using Machine Learning (Data Analytics),” Master’s Thesis, University of Tartu, 2019.

[42] V. E. Lee *et al.*, “Does cognitive performance predict contest outcome in pigs?,” *Anim. Behav.*, vol. 214, pp. 27–41, 2024.

[43] X. Dopico-Calvo *et al.*, “The penalties and scores by events, to predict victory and defeat according to when the contest ends in elite judo contests,” *Rev. Artes Marciales Asiáticas*, vol. 19, no. 2, pp. 93–103, 2024.

[44] F. Shull, J. Singer, and D. I. K. Sjøberg, *Guide to Advanced Empirical Software Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.