AROhI: An Interactive Toolkit for Estimating ROI of Data Analytics

Noopur Zambare[*], Jacob Idoko[†], Jagrit Acharya[†], and Gouri Ginde[†]

[*]*Dept. of Mechanical Engineering, Indian Institute of Technology Jodhpur, India*
Email: zambare.1@iitj.ac.in
[†]*Dept. of Electrical and Software Engineering, University of Calgary, Canada*
Email: {jacob.idoko, jagrit.acharya1, gouri.ginde}@ucalgary.ca

Abstract—The cost of adopting new technology is rarely analyzed and discussed; however, it is vital for any technological investment because of the cost and effort associated with it. Thus, it is crucial to consider Return On Investment (ROI) when performing data analytics. Decisions on "How much analytics is needed?" are hard to answer. ROI could guide decision support on the What?, How?, and How Much? Analytics for a given problem. This work details a comprehensive tool that provides conventional and advanced ML approaches for demonstration using software requirements dependency extraction and their ROI analysis as a use case. Utilizing advanced ML techniques such as Active Learning, Transfer Learning and primitive Large language model: BERT (Bidirectional Encoder Representations from Transformers) as its various components for automating dependency extraction, the tool outcomes demonstrate a mechanism to compute the ROI of ML algorithms to present trade-offs between the cost and benefits of a technology investment.

Index Terms—Data Analysis, Return on Investment, Machine Learning

I. INTRODUCTION

Most Machine Learning (ML) solutions are evaluated based on the algorithm's performance measures, such as accuracy and AUC-ROC (Area Under the Receiver Operating Characteristic) Curve, precision and recall. However, ML algorithms for a particular problem typically consume more computational resources than classical algorithms for the same problem. Furthermore, these methods require large training (annotated) datasets, which tends to become an effort and cost-intensive aspect. Hence, decisions on "How much is needed?" are hard to answer; however, ROI could be used as guidance for decision support in deciding What? How? and How Much? of analytics for a given problem [1].

Our contributions in this work are as follows.

- To over knowledge, we are the first to develop a cloud hosted tool that enables comparison and evaluation of the ML approaches not only using ML performance measures but also based on ROI criteria.
- Our comprehensive interactive tool has been designed to visualize the ROI and ML performance measurement trade-offs, thus providing a decision support for the technological investment decision.
- Demonstrate how users can change cost factors and view the projections of change ROI for decision-making through an interactive dashboard-like interface.

- We demonstrate the utility of the solution application through the software requirements dependency extraction use case for a dataset mined from open source web-browsing application, Firefox, from a Mozilla family of projects.

Need for ROI estimation: In most ML projects, accuracy is often the most reported metric for evaluating the performance of a model. However, this can be misleading and result in a fallacy that undermines the actual value of a model.

- The accuracy fallacy: The "accuracy fallacy" is that a model's high accuracy equates to reliability and performance, which is rarely true. Accuracy measures how often a model's predictions are correct. However, it does not account for the cost of errors, such as false positives and false negatives. For instance, a model that predicts rare events can achieve high accuracy by simply predicting the majority class, which does not provide meaningful insight into the minority class. Accuracy also fails to account for the balance between recall (the ability to find all relevant instances) and precision (the ability to find relevant instances) [2]. In most cases, recall should carry more weight than precision, as noted by Berry et al. [3] in their empirical evaluation of tools for requirements engineering tasks. These limitations call for more robust evaluation metrics that align with business objectives.

To truly evaluate a model's effectiveness, it is necessary to estimate its ROI. This involves considering factors such as lift, which measures how much better the model is compared to random guessing, and the specific costs associated with different types of errors [4]. In summary, it captures the multiplicative improvement to operations and the increased return for your efforts, i.e., ROI. Wafiq Syed, a Data Product Manager at Walmart [4], notes, "The most important metric for your model's performance is the business metric that it is supposed to influence". Metrics such as accuracy reports can answer, "How often is the model correct?" for both positive and negative cases, make a bad model look good. For example, if only 1 percent of the customers buy, a model that predicts "no" for every customer achieves 99% accuracy but fails to predict a potential customer correctly.

-Evaluating ML solutions beyond performance: While there is a broad spectrum of ML techniques for task automation, not all techniques are economically viable in all the scenarios, considering the size of training data and the investment in effort. When in doubt, simple ML techniques

are recommended over complex ones [5], [6]. However, what if a method existed to estimate such investments and compute the benefits when other complex algorithms are chosen? Interestingly, scholarly research exploring the ROI of ML is scarce [7] [8]. However, a few discussions and approaches have been proposed in the gray literature, especially on the business front, to weigh the benefits of AI/ML-based software solutions in terms of economic value (dollars). For instance, [9] emphasizes the need for ROI as a measure for evaluating the benefits of ML. Conversly, in another blog [10], a method is proposed to arrive at ROI economic value, which emphasizes the cost incurred due to prediction outcomes of the ML technique. Although these approaches are intriguing and align well with our agenda, they need more empirical rigour and evaluation. They do not consider the cost of data accumulation and pre-processing, a large part of applied ML effort [11].

Hence, ML techniques must be evaluated beyond performance measures for effective decision-making. In other words, it is necessary to understand how far one should chase the accuracy parameter when the cost of achieving the differential is substantial.

II. RELATED WORK

In recent years, the assessment of the ROI for ML solutions across various domains has increased. This section explores existing research, tools, and frameworks, focusing on ROI estimation in data analytics, particularly for ML and AI applications.

Existing frameworks for measuring the business value of Data Analytics: Several frameworks have been developed to enhance ROI estimation within data analytics and ML projects. These tools provide methodologies to estimate the financial impact and value of AI/ML-driven solutions. Some notable examples among them include the following.

- *Net Present Value (NPV):* NPV calculates the present value of a data analytics project's expected future cash flows. It considers the initial investment, expected returns, and the time value of money [12] [13].
- *Cost-Benefit Analysis (CBA):* CBA compares the costs associated with implementing and maintaining a data analytics project with the benefits it generates. It considers tangible and intangible costs and benefits to determine the overall ROI [14].
- *Return on Investment Ratio*: A simple ratio that compares the net benefits of a data analytics project to its costs. It is expressed as a percentage and provides a quick snapshot of the project's financial performance [15].
- *Break-Even Analysis:* This model determines the point at which the benefits of a data analytics project equal its costs. It helps identify the time it takes for the project to reach a break-even point and start generating positive returns [16].
- *Value-Based ROI:* This framework focuses on measuring the value created by a data analytics project in terms of its impact on key performance indicators (KPIs) or strategic objectives. It goes beyond financial metrics to assess the project's overall value proposition [17].

- *Total Cost of Ownership (TCO):* TCO considers the total lifecycle costs of a data analytics project, including upfront investment, operational costs, maintenance, and potential upgrades. It provides a comprehensive view of the project's financial impact [18].

Industry applications of ROI: ROI estimation tools and frameworks are commonly used in various sectors to improve decision-making processes and boost business outcomes with the help of data analysis and ML. Some specific uses in the industry include:

- *Finance:* ROI estimating tools and frameworks play an important role in finance by optimizing investment strategies and predicting trends in the market. Lee et al. [19] showed that AI models such as LSTM improve decision-making in real estate investments and reveal high ROI [19]. Net Present Value (NPV) and ROI ratio are used to measure AI/ML-based investment strategies' financial feasibility and performance [20].
- *Healthcare:* Healthcare is a field that AI/ML and data analytics have greatly influenced in enhancing patients' quality of life. Pandey et al. [21] highlight AI/ML's impact on enhancing ROI by improving the efficiency of operations while at the same time reducing the costs involved [21]. Cost Benefit Analysis (CBA) analyzes the economic benefits of healthcare performance indicators such as patient recovery rates and the use of resources [22].
- *Manufacturing and Aerospace:* AI/ML and robotics integration in manufacturing and aerospace increases productivity and operational efficiency, thus delivering rapid ROI. Bertoni et al. [23] showed that integrating value and sustainability assessments in the exploration of design spaces through ML optimizes design processes and reduces costs. In addition, Figy [24] discussed how Labplas achieved an ROI of less than one year by redesigning custom production machines with EtherCAT, robotics and AI for quality control. This resulted in an increase of 15% to 35% in machine productivity [24]. Break-even analysis is used to identify the point at which the benefits of the new technology equal its costs. At the same time, Total Cost Ownership (TCO) evaluates the full lifecycle costs against the gains in productivity.

However, to our knowledge, no tool has been developed yet that can be used to determine the value of data analytics through a no-code interaction interface. This study is a step towards achieving such a feat.

III. THE AROHI TOOL

The AROhI (Data **A**nalytics and its **R**eturn **O**n h**I**nvestment)[1] tool provides users with a cloud-deployed dashboard, enabling estimation of the ROI of Data Analytics. This tool is a continuation of the work from [1] [25] and [26]. **This no-code tool** enables utilizing ML in technology investment-based decision-making through advanced data

---

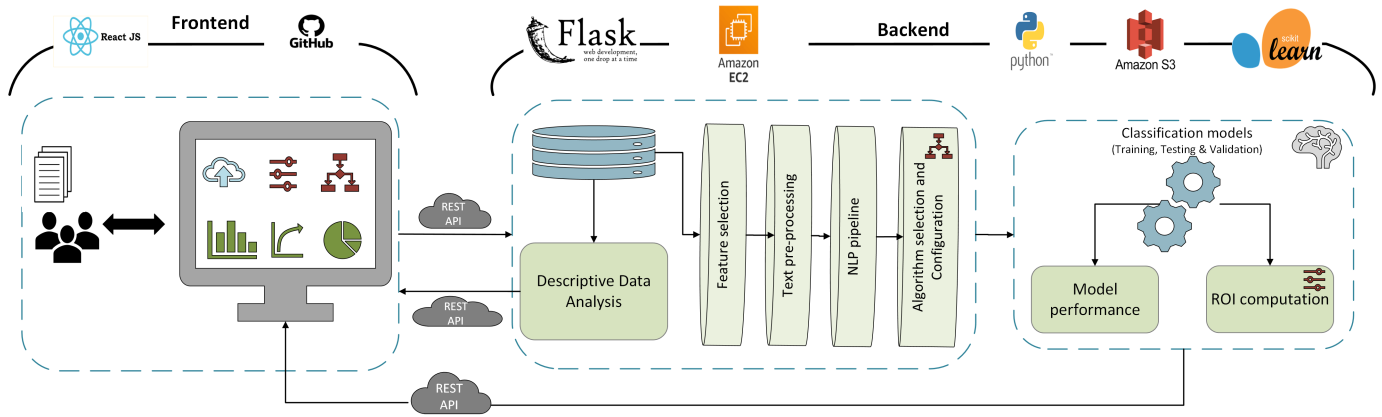[1] AROHI also means something which/who is always striving for something more in Sanskrit language

Fig. 1. Conceptual design of the AROhI tool and various technology components of it

analysis and ROI estimations, thus empowering non-programmer decision-makers to create and customize ML-driven analytics. The detailed working of the tool has been captured in a demonstration video through the anonymous link to folder[2].

Architecture: Figure 1 shows the conceptual design of this tool with various technology components used in its development, and figure 2 shows its user interface and capabilities. AROhI has a front-end user interface that was developed using the ReactJs framework. The interactive interface allows end users to upload the dataset and provide additional settings on algorithm selection and cost factors for ROI computation. REST APIs communicate with the backend server developed in the Flask framework. Communication between front end (UI) and back end are managed through REST APIs. The backend server was developed using Python programming language, utilizing the Amazon S3 database for data storage. ML algorithms are developed using SciKit Learn, an open-source software library.

ML Algorithms: AROhI comprises supervised learning algorithms, including Logistic Regression, naive Bayes, Decision Trees, Support Vector Classifier (SVC), Random Forest Classifier, and fine-tuned BERT. The tool also provides an option to use semi-supervised learning, such as Active learning. The user can modify several settings of the active learning algorithm, including the threshold, maximum iteration, test size, sampling type, classifier type, resampling, and manual annotations count. Active learning [27] attempts to maximize a model's performance gain while annotating the fewest samples possible [28]. With increasing data, the demand and computational cost of data analysis using ML algorithms have gradually risen. Hence, considering both accuracy and investments, it is important to consider a holistic approach to evaluate these algorithms. This approach considers broader business impacts, thus offering a more thorough evaluation of costs and benefits for data analysis.

Features and Capabilities: Descriptive analysis of the input dataset is shown in the front end (UI) utilizing various charts and graphs (visualization: discussed in use case Section IV). The data set is further preprocessed through a data pipeline (❷ of Figure 2) enables the end user to observe the features present within the data and further drill down through interactive charts to visualize statistical information through bar graphs. Users can further choose the desired dependent and independent variables (features).

As shown in the flow diagram of in Figure 3, AROhI provides an option for the end user to procure additional annotations through employing semi-supervised ML such as Active Learning [27] if the dataset is smaller or utilizing advanced ML methods such as Fine-tuned BERT (currently implementation is work in progress) over conventional ML algorithms (discussed in this work).

Furthermore, the dataset is processed through a natural language processing (NLP) pipeline. This pipeline includes lemmatization (a process that uses vocabulary and morphological analysis of words to remove inflectional endings and return words to their dictionary form) [29], and vectorization (a technique in NLP to map vocabulary words to a vector of real numbers.) as part of the text processing.

The processed data goes to the ML Algorithms during the next stage. Training and testing subsets are created based on the training size provided by the user. We have designed two learning techniques, including supervised and semi-supervised algorithms. In supervised learning, annotated data is used to predict the values of the output [30] [31]. Most of the time, the use cases need more sufficient labelled data. Often, labelling data is expensive where unlabeled data is available in abundance [31]. Semi-supervised learning addresses these issues and works better than supervised learning [31].

ML models are evaluated based on their accuracy, 10-fold cross-validation score, confusion matrix and F1 score. Additionally, precision and recall are also shown using incremental train set size. Beginning with the 20% of the training

---

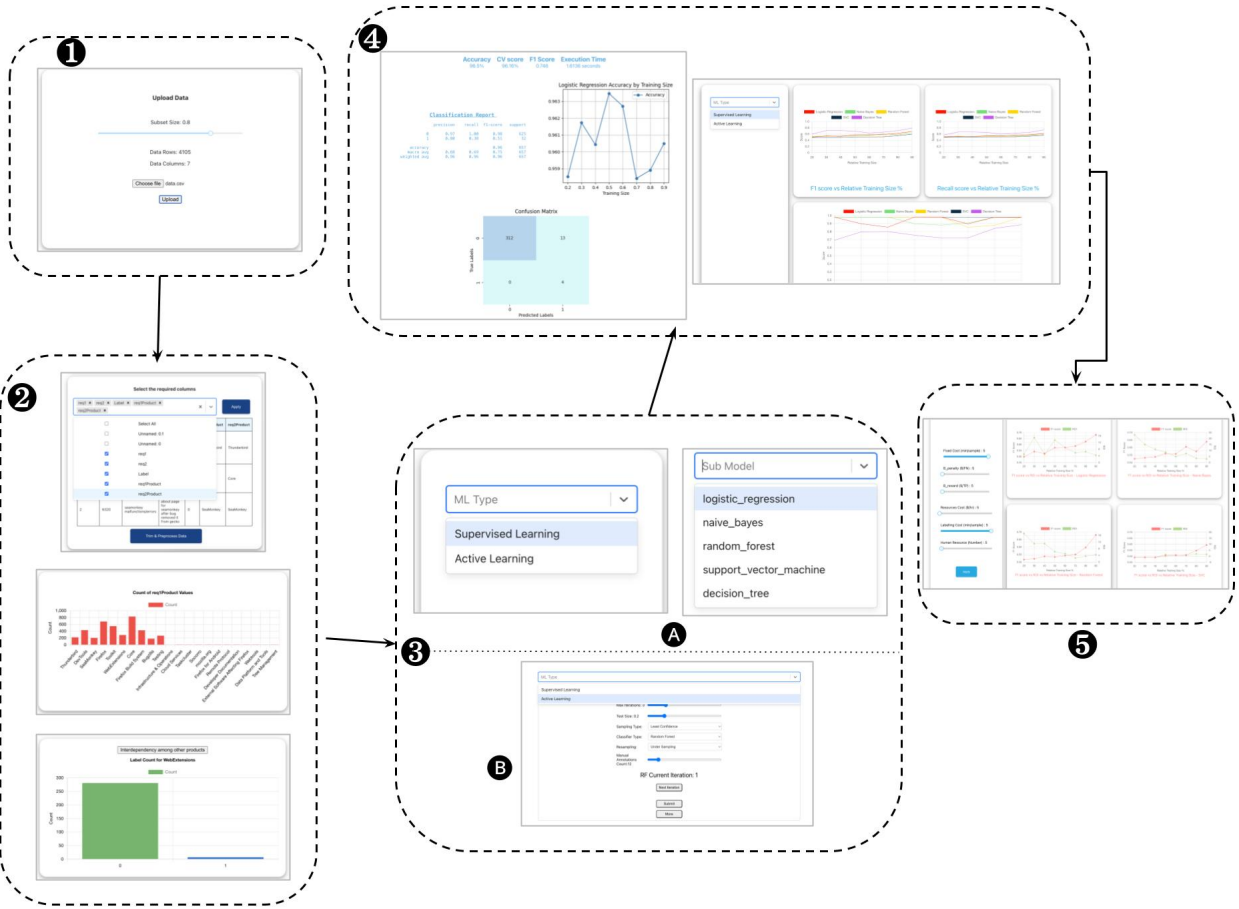[2]https://anonymous.4open.science/r/ROI_tool-1DC8/README.md

Fig. 2. AROhI Tool's user Interface and various screens of the tool enable users to interact and draw conclusions
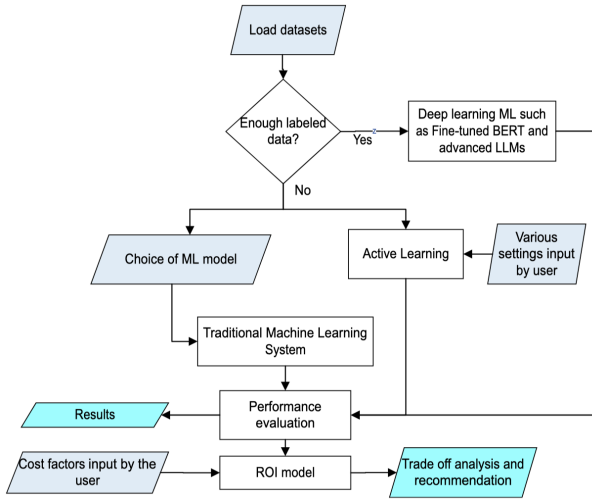

Fig. 3. Overall workflow diagram of AROhI tool: Enables end user to choose the desired supervised or semi-supervised method to evaluate its usefulness for their problem and ROI analysis

set, models are developed to capture their performance by incrementally growing (increasing the train by 10%) training set size to capture its impact on F1 accuracy and ROI. Users

can monitor variations in accuracy when training sizes are changed, as the classification matrices are provided to help users better understand their model. The user can also compare all the models based on F1 score, recall and precision in the user interface.

For ROI computations, end users are provided with slider bars to choose various cost factors (discussed in Section III) so that the incremental (with increasing train set) ROI estimates corresponding to the performance of the chosen models are determined. Using sliders, interaction with cost factors config-uration generates dynamic graphs for comparative ROI and F1 score analysis for varying training sizes based on user input. The generated graphs are visualized in the user interface as shown in Figure 1.

The relationship between the framework of AROhI in Figure 1 and the AROhI software (tool) in Figure 2 is as follows:

- In Figure 1, the first component ❶ corresponds the figure 2, depicting data import. Next, the data analysis is performed in component ❷ of Figure 2.
- Through REST APIs, the front end communicates with the backend shown in component ❷ of the figure 1. Data passes through the NLP pipeline, leading to the process in component ❸ of Figure 2, where ML model selection is

made. A comparison of various models built using scikit-learn is shown in component ❹.

- In the last component of Figure 1 where ROI computations and configurations of costs happen corresponds to the component ❺ of Figure 2. Finally, the results are sent to the front end (UI) using REST APIs.

	Parameters	Symbol	Unit
Cost	Data gathering time	C_{dg}	Minutes
	Pre-processing time	C_{pp}	Minutes
	Evaluation time	C_e	Minutes
	Labeling time	C_l	Minutes
	Resource Cost	$C_{resource}$	$ per hour
Benefit	Value per True Positive instance	B_{reward}	$
	Penalty per False Negative instance	$B_{penalty}$	$
Other	H	Human resources	Number
	N_{train}	Size of the training set	Number
	N_{test}	Size of the testing set	Number
	N	$N_{train} + N_{test}$	Number

ROI Modeling: Following a typical approach to the ML life cycle, Planning, Data Preparation, Execution and Validation are determined as various phases to further associate cost factors with each one as shown in Table I. Table II depicts the parameter settings used. These parameters and settings are derived from the work by [1]. Users can configure fixed costs, labelling costs, resource costs, reward per true positive instance, penalty per false negative instance, and the number of human resources used. The fixed cost and labelling cost together form the data preprocessing cost [1].

To demonstrate cost-effectiveness, we focused on financial metrics such as ROI, which is similar to the approach used to evaluate the profitability of software based on real-world cases [32] [1].

The ROI formula is given by:

$$\text{ROI} = \frac{\text{Benefit} - \text{Cost}}{\text{Cost}} \quad (1)$$

The benefit calculation formula is given by:

$$\text{Benefit} = TP \times B_{reward} - FN \times B_{penalty} \quad (2)$$

The cost calculations are made by using:

$$\text{Cost} = N\% \times \frac{(C_{fixed} + C_l)}{60} \times H \times C_{resource} \quad (3)$$

Unit for benefit and cost is the human effort in person-hour [25].
Evaluation Metrics: We are letting users evaluate their results using the following metrics:

- **Cross validation:** It is statistical method of evaluating Furthermore, comparing learning algorithms by dividing data into two segments: one used to learn or train a model

Parameters	Value
$C_{fixed} = C_{dg} + C_{pp} + C_e$	1 min/sample
C_l	0.5 min/sample
$C_{resource}$	$400/hr
B_{reward}	$500/TP
$B_{penalty}$	$500/FN
H	1
N	4105

and the other used to validate the model [33]. We have used k-fold cross-validation with k = 10.
- **Confusion Metrics:** We obtained a 2×2 matrix for each model indicating Actual Negative, Actual Positive, Predicted Negative and Predicted Positive [25].
- **F1 Score:** It is calculated using actual and predicted classes.

$$\text{F1 score} = \frac{(2 \times TP)}{2 \times TP + FP + FN} \quad (4)$$

- **ROI:** As shown in Equation (1), it is a financial metric used to evaluate the profitability of an investment.

IV. USE CASE

To demonstrate the utility of this tool, we utilize the requirements dependency extraction problem for the dataset from the Mozilla family of products [34] as a use case (such as Firefox browser, Thunderbird email client, etc.). A requirement is a condition or capability a user needs to solve a problem or achieve an objective. It is also a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents [35]. To demonstrate the utility of our solution, we performed binary classification of requirements dependency types using a subset of the entire dataset for training and testing purposes. This process was iterated by gradually increasing the proportion of the training set and storing the results for further calculations. **This allows users to** understand the trade-off between the F1 score and ROI and the eventual break-even point of positive returns.

Dataset: Bugzilla is a popular online bug-tracking systems utilized in open-source software development [34]. Mozilla family of projects utilize this system to track its bugs [34]. For this study, data from Firefox, an open-source Mozilla Web browser application, originally mined and prepared by ¡redacted for blind review¿ et al. [26]. The 4,105 samples used in this study belonged to the Firefox project and Table III represents pairs of requirements dependencies from this dataset.

Given a set of requirements R and a pair of requirements $(r, s) \in R \times R$, we consider two requirements INDEPENDENT if addressing one of them does not have any logical or practical implications for addressing the other. Otherwise, they are labeled DEPENDENT [26].

TABLE III
SAMPLES OF DEPENDENCY PAIRS FROM OUR USE CASE DATASET.

Dependency type	Requirement 1	Requirement 2
DEPENDENT	implement quantum bar design update behind pref	shift bookmark toolbar downwards when urlbar is focused in new tab
	update navigation toolbar overflow panel for photon	make bookmark toolbar item blend in better in overflow panel
INDEPENDENT	offer choice to open natively with content disposition attachment view/show in browser	remove old onboarding add on prefs
	context menu option to open link/image using external application	add automated test for item from bookmark toolbar can be sorted by name

Execution The dataset is processed following the workflow in 1 and 3.

- To select the initial subset of the data, we used its 80%. We kept 20% for testing and validation from the subset of training data depicted in ❶ of Figure 2.
- component ❷ of Figure 2 shows that requirements and their descriptions were selected during feature selection. Figure 1 shows the tool's workflow when the user inputs the data through front end (UI). However, users can customize the selection using the dropdown in the tool. The chart is displayed on the front end to determine the characteristics and quantify the number of various requirements in the data.
- In the next component, the user can compare the model results for semi-supervised shown in ❸b and supervised learning in ❸a using different classifiers.
- In Active Learning, the user can choose from available sampling techniques, threshold, resampling technique, manual annotation count and classifier type. Component ❸ of Figure 2 illustrates, how the ML component of the tool works. This component stores the true positive and false negative count. Next is to compare different models regarding F1 score, recall and precision shown in component ❹, Figure 2. The final component in ❺ is to calculate benefit and cost using the data the user selects through slider and ML models stored data.

V. RESULTS

Table V shows the accuracies of various supervised learning models where execution time denotes the time taken to execute the process from triggering the model for training for the initial set training size defined by the user (0.8 in this case). We also performed the 10-fold cross-validation during training of this initially set data. Table IV shows the variation of F1 score and ROI with training data size.

For Logistic Regression: The best accuracy can be reached with a dataset of up to 60%. Beyond that, the ROI drops. Although the highest accuracy is with 90% of the dataset, the ROI drops sharply with the given settings:

Naive Bayes: The highest ROI is with just 20% of the dataset, so investing more is not useful. although NB's performance becomes comparable with others at a certain point, it does not yield value beyond 40% at .54 accuracy. Thus, NB might not be a good algo for the given problem.

Random Forest: Accuracy does not grow until 80% of the train set however, similar to NB, ROI is maximum at just 20% of the dataset, so even RF might not be a good algorithm to choose from.

SVC: Although ROI is positive only after 50% of the dataset is procured for training. Also, the accuracy is poor, and the algorithm struggles to understand the dataset's underlying pattern, making this one an inferior choice. Also, beyond 80%, the ROI drastically decreases while accuracy increases marginally. Also, it shows that the break-even is achieved only after investing in 40% training data acquisition.

Decision Tree: Achieves best accuracy of .72 just with 20% of the dataset with the highest ROI, making this one the most valuable and effective algorithms for the given problem and values.

ROI Sensitivity Analysis As an outlook to varying input costs, we envision enhancing the tool further to show how ROI varies by varying the various cost factors from Table II. For example, we fixed the training size to 80% for Logistic Regression under supervised learning and then increased Resource Cost from $400 to $440 (10% increase), and ROI decreased from 389.62 to 354.11 (9.11% drop). Enabling such visualization remains part of the AROhI tool's feature list. As show in Figures 4, 5, 6, and 7 to visualize such analysis dynamically.

VI. LIMITATIONS

This research has a few limitations: (i) The use case evaluation is on a dataset that uses less intense computational resources. Rigorous testing of the tool remains part of our future work. However, we are confident that the results are generalizable due to the research finding in the related work [1], [25]. (ii) The tool currently supports classification problems; Extending to other problem types remains part of our future work. (iii) The tool is trained on independent and dependent types of requirements, which we plan to extend to other multi-class classification problems in the future. (iv) Other aspects of the ML, such as bias, fairness, and carbon footprint, have implications for ROI. Thus, instead of evaluating the models based on performance measures and ROI alone, we consider these aspects while computing ROI. However, this remains part of our future work.

TABLE IV
F1 AND ROI FOR THE SETTINGS IN TABLE II

| Training size	Logistic regression		Naïve Bayes		Random Forest		SVM		Decision Tree	
	F1	ROI	F1	ROI	F1	ROI	F1	ROI	F1	ROI
20%	0.50	311.50	0.52	**1249.00**	0.52	936.50	0.49	-1.00	0.60	-938.50
30%	0.54	832.33	0.53	832.33	0.51	415.67	0.49	-1.00	**0.72**	**2082.33**
40%	0.53	311.50	0.54	624.00	0.54	**467.75**	0.49	-1.00	0.70	1717.75
50%	0.57	**749.00**	0.56	499.00	0.53	**374.00**	0.50	**124.00**	0.69	**749.00**
60%	0.58	519.83	0.56	311.50	0.54	**311.50**	**0.51**	103.17	0.64	-1.00
70%	0.59	**356.14**	0.60	445.43	0.55	**266.86**	**0.51**	88.29	**0.68**	-90.29
80%	0.62	**389.63**	0.57	**233.38**	0.60	**311.50**	0.55	**155.25**	0.68	311.50
90%	**0.68**	**276.78**	**0.64**	**207.33**	**0.68**	**276.78**	**0.59**	137.89	**0.72**	**207.33**
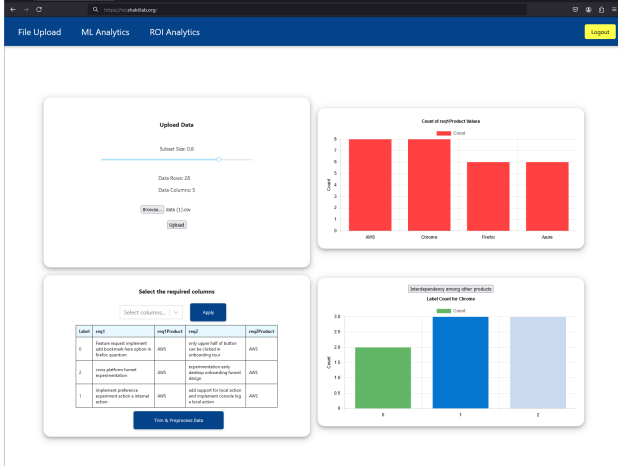
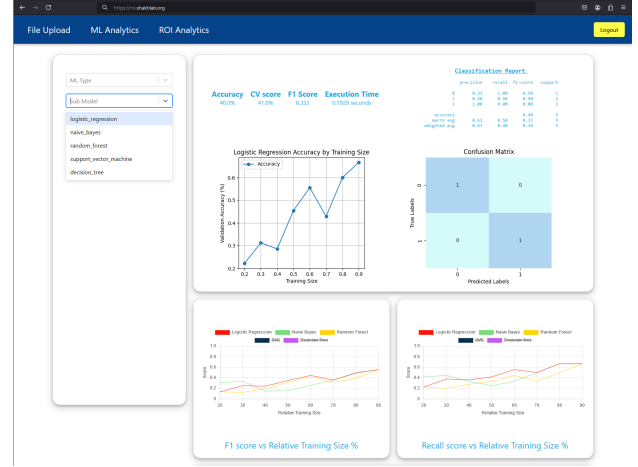Fig. 4. The login based UI shows descriptive analysis of the uploaded data

Fig. 5. Various graphs showing relative train set and various ML measures such as F1-Score, Precision and Recall
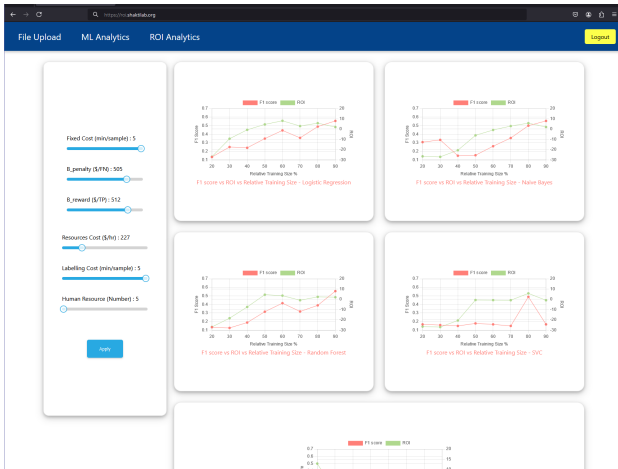

Fig. 6. For the chosen cost factors, ROI vs F1 score charts are generated dynamically showing at what point break even is achieved and how the ROI varies with increasing train set for ML algorithms
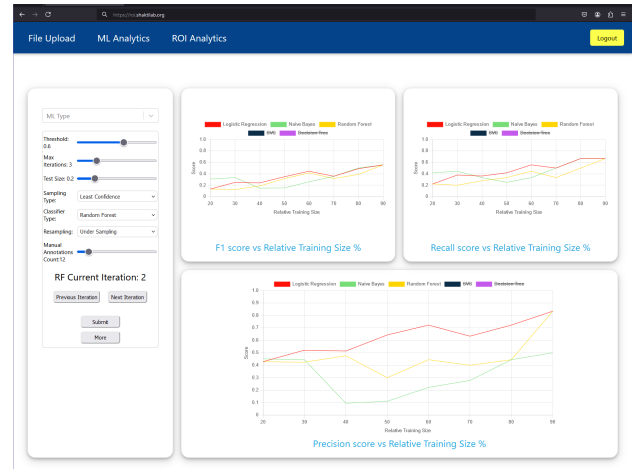

Fig. 7. For various settings using the sliders on the left side, Active Learning based algorithms can be used to accumulate additional annotations incrementally. This is work in progress

VII. CONCLUSION AND FUTURE WORK

In this study, we propose the AROhI tool to compute the ROI of data analytics and its trade-off analysis with ML performance metrics such as precision, recall and F1-score, thus enabling end users (stakeholders and decision-makers) to decide on technological investment. This tool provides an extensive and interactive interface to analyze and compare various data analysis techniques using ML based on accuracy, costs and benefits, thus providing a holistic ROI-driven approach to any classification problem. In our future

TABLE V
ACCURACY OF SUPERVISED LEARNING MODELS FIXING TRAIN SIZE = 0.8

Model	Accuracy %	CV score %	Execution Time (sec)
Logistic Regression	96.50	96.16	1.60
Decision Tree	94.82	95.95	2.36
Naive Bayes	92.39	96.07	0.781
SVC	95.89	96.07	9.35
Random Forest	95.28	96.16	21.79

work, we aim to improve the performance and applicability of our tool across a broader range of datasets, domains and advanced AI/ML methods such as large language models (LLMs). Further, we also plan to address the following areas:

1) We intend to generalize our approach to other domains, such as health and manufacturing, where applied AI/ML have scope for utilization. Also, we will modify the ROI model to incorporate cost and benefit investments for Generative AI-related problems and solutions. This will ensure that our model is versatile and adaptable to different problems and AI methods.

2) In addition, we will apply unsupervised learning algorithms, improving the model's ability to detect underlying patterns without relying on labeled data.

3) Along with ROI and accuracy, we will introduce additional metric to evaluate bias and fairness and its trade-off with the investment made in technology.

REFERENCES

[1] R. for double blind review, "Beyond accuracy: Roi-driven data analytics of empirical data," in *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–6, 2020.

[2] J. DiMatteo, D. M. Berry, and K. Czarnecki, "Requirements for monitoring inattention of the responsible human in an autonomous vehicle: The recall and precision tradeoff.," in *REFSQ Workshops*, 2020.

[3] D. M. Berry, J. Cleland-Huang, A. Ferrari, W. Maalej, J. Mylopoulos, and D. Zowghi, "Panel: Context-dependent evaluation of tools for nl re tasks: Recall vs. precision, and beyond," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 570–573, 2017.

[4] E. Siegel, *The AI Playbook: Mastering the Rare Art of Machine Learning Deployment*. MIT Press, 2024.

[5] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[6] C. Bird, T. Menzies, and T. Zimmermann, *The art and science of analyzing software data*. Elsevier, 2015.

[7] S. Nagrecha and N. V. Chawla, "Quantifying decision making for data science: from data acquisition to modeling," *EPJ Data Science*, vol. 5, no. 1, p. 27, 2016.

[8] M. Ferrari *et al.*, "Roi in text mining projects," *WIT Transactions on State-of-the-art in Science and Engineering*, vol. 17, 2005.

[9] www.altexsoft.com, "Measuring roi for machine learning and data science." https://www.altexsoft.com/blog/business/how-to-estimate-roi-and-costs-for-machine-learning-and-data-science-projects/. Accessed: July 2024.

[10] "Roi of machine learning." https://towardsdatascience.com/return-on-investment-for-machine-learning-1a0c431509e#:~:text=Just%20like%20with%20any%20investment,cost%20of%20mistakes%20and%20accuracy. Accessed: July 2024.

[11] I. Figalist, C. Elsner, J. Bosch, *et al.*, "An end-to-end framework for productive use of machine learning in software analytics and business intelligence solutions," in *Inter. Conf. on Product-Focused Software Process Improvement*, pp. 217–233, Springer, 2020.

[12] E. F. Brigham, *Financial management: Theory and practice*. Cengage Learning Canada Inc, 2016.

[13] T. Koller, M. Goedhart, and D. Wessels, *Valuation: measuring and managing the value of companies*. John Wiley & Sons, 2010.

[14] D. Baracskay, "Cost-benefit analysis: Concepts and practice," 1998.

[15] J. J. Phillips, *Return on investment in training and performance improvement programs*. Routledge, 2012.

[16] J. G. Siegel, J. K. Shim, and S. Hartman, "Schaum's quick guide to business formulas: 201 decision-making tools for business, finance, and accounting students," *(No Title)*, 1998.

[17] G. B. Stewart, "The quest for value: a guide for senior managers," *(No Title)*, 1991.

[18] L. M. Ellram, "Total cost of ownership: an analysis approach for purchasing," *International Journal of Physical Distribution & Logistics Management*, vol. 25, no. 8, pp. 4–23, 1995.

[19] J.-h. Lee, H.-b. Kim, and G.-e. Shim, "Prediction model of real estate roi with the lstm model based on ai and bigdata," *International journal of advanced smart convergence*, vol. 11, no. 1, pp. 19–27, 2022.

[20] Ginimachine, "Understanding the roi of implementing ai in financial services." https://ginimachine.com/blog/the-roi-of-implementing-ai-in-financial-services/, 2024.

[21] S. Pandey, S. Gupta, and S. Chhajed, "Roi of ai: Effectiveness and measurement," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume*, vol. 10, 2021.

[22] R. J. Brent, "Cost-benefit analysis versus cost-effectiveness analysis from a societal perspective in healthcare," *International Journal of Environmental Research and Public Health*, vol. 20, no. 5, p. 4637, 2023.

[23] A. Bertoni, S. K. Dasari, S. I. Hallstedt, P. Andersson, *et al.*, "Model-based decision support for value and sustainability assessment: Applying machine learning in aerospace product development," in *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 2585–2596, 2018.

[24] J. Figy, "Pc-based control boosts sterile sampling bag production," *Control Engineering*, vol. 68, no. 2, pp. 22–25, 2021.

[25] R. for double blind review, "Is bert the new silver bullet?-an empirical investigation of requirements dependency classification," in *2021 IEEE 29th international requirements engineering conference workshops (REW)*, pp. 136–145, IEEE, 2021.

[26] R. for blind review, "How much data analytics is enough? the roi of machine learning classification and its application to requirements dependency classification," *arXiv preprint arXiv:2109.14097*, 2021.

[27] B. Settles, "Active learning literature survey," 2009.

[28] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[29] L. Zhao, W. Alhoshan, A. Ferrari, and K. J. Letsholo, "Classification of natural language processing techniques for requirements engineering," *arXiv preprint arXiv:2204.04282*, 2022.

[30] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," *The elements of statistical learning: Data mining, inference, and prediction*, pp. 9–41, 2009.

[31] Y. Reddy, P. Viswanath, and B. E. Reddy, "Semi-supervised learning: A brief review," *Int. J. Eng. Technol*, vol. 7, no. 1.8, p. 81, 2018.

[32] S. Stradowski and L. Madeyski, "Costs and benefits of machine learning software defect prediction: Industrial case study," 2024.

[33] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of database systems*, pp. 532–538, 2009.

[34] Mozilla.org, "Bugzilla: bug-tracking system." https://www.bugzilla.org/, Apr. 2020.

[35] "Ieee standard glossary of software engineering terminology," *ANSI/IEEE Std 729-1983*, pp. 1–40, 1983.