# 

# **1. Title Page**

**Report Title:**

**Comparative Analysis of Kickstarter and Indigo Crowdfunding Data Using Machine Learning Algorithms.**

**Author:  
Kunal Punjabi**

**Affiliation:**Vivekanand Education Society Institute of Technology  
Department of Information Technology

**Supervisor:**Mrs. Kajal Jewani

**Submission Date:**10-12-2025

## **2. Abstract**

Crowdfunding platforms such as Kickstarter and Indiegogo have transformed how creative projects, startups, and innovations raise capital. However, the success of a campaign is far from guaranteed and depends on multiple factors including funding goals, promotion, category, and timing. This study presents a comparative analysis of Kickstarter and Indiegogo crowdfunding performance using machine learning algorithms, with a focus on predictive modeling and risk analysis for new startup campaigns.

Two cleaned datasets were used: an Indiegogo dataset of approximately **121,605 rows and 20 features**, and a Kickstarter dataset of **100,000 rows and 30 features**. After preprocessing and feature engineering, multiple classification algorithms were trained and evaluated, including **Random Forest, XGBoost, and AdaBoost**, along with ensemble techniques such as **soft voting** and **hard voting**. Hyperparameter tuning was performed using **RandomizedSearchCV** with cross-validation to optimize model performance.

On Indiegogo data, a three-model soft voting ensemble combining Random Forest, XGBoost, and AdaBoost achieved **100% accuracy**, significantly improving over the initial baseline of 90.16%. On Kickstarter data, a tuned **XGBoost model** achieved a best cross-validated accuracy of approximately **82.22%**, outperforming both the baseline Random Forest (79.03%) and the voting ensembles.

The analysis identifies key success-driving parameters such as **funds raised percentage, goal size, promotion, and prototype readiness**, and proposes a **risk analysis framework based on goal size categories** (Very Low, Low, Medium, High, Very High) for assessing the risk level of new startup campaigns. The results demonstrate that tuned gradient boosting models can provide robust and interpretable predictions and that risk analysis based on funding goal and platform behavior can guide more informed decision-making for campaign planning.

## **3. Introduction**

Crowdfunding platforms have emerged as powerful tools for financing new products, creative endeavors, and early-stage startups. Platforms such as Kickstarter and Indiegogo allow project creators to present their ideas to a global audience and collect funds in relatively small contributions from many backers. While this model democratizes access to capital, it also introduces high uncertainty: many campaigns fail to reach their funding goals, resulting in lost effort and opportunity.

In this context, **crowdfunding analytics** and **predictive modeling** play a crucial role. By learning from historical campaign data, machine learning models can predict whether a new campaign is likely to succeed or fail, and can highlight the key parameters affecting success. Such insights are valuable not only for project creators, but also for platforms and investors who wish to assess campaign risk.

This study focuses on two of the most popular platforms:

* **Kickstarter** – widely used for creative and technology projects.
* **Indiegogo** – known for innovation, flexible funding, and startup-focused campaigns.

Both platforms provide rich historical data, including information about funding goal, pledged amount, backers, category, timing, and other status flags.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **Objectives**

The main objectives of this study are:

1. **Platform-wise comparison**
   * Compare predictive model performance on Kickstarter vs. Indiegogo datasets.
2. **Algorithm evaluation**
   * Evaluate **Random Forest, XGBoost, and AdaBoost**, and ensembles (soft/hard voting).
3. **Impact of hyperparameter tuning**
   * Assess how tuning improves accuracy and stability over baseline models.
4. **Success parameter identification**
   * Identify which features (e.g., goal amount, percent funded, promotion, duration) contribute most to campaign success on both platforms.
5. **Risk analysis for new startups**
   * Propose a simple but practical **risk analysis framework** based on goal size categories and success probabilities, which a new startup can use to estimate its risk level before going live.

## **4. Literature Review**

Several prior studies have examined crowdfunding platforms to understand what drives campaign success:

* Research has shown that **funding goal**, **project category**, **video presence**, **social proof** (number of backers, comments), and **campaign updates** have a strong influence on success probability.
* Machine learning models such as **Logistic Regression, Random Forest, Gradient Boosting, and XGBoost** have been widely used to predict campaign outcomes.
* Some studies have integrated **text analytics** (project descriptions, titles) using NLP methods such as TF–IDF and sentiment analysis to further improve predictions.
* **Imbalance handling techniques** like SMOTE and cost-sensitive learning are also frequently reported due to the uneven distribution of successful vs failed projects.

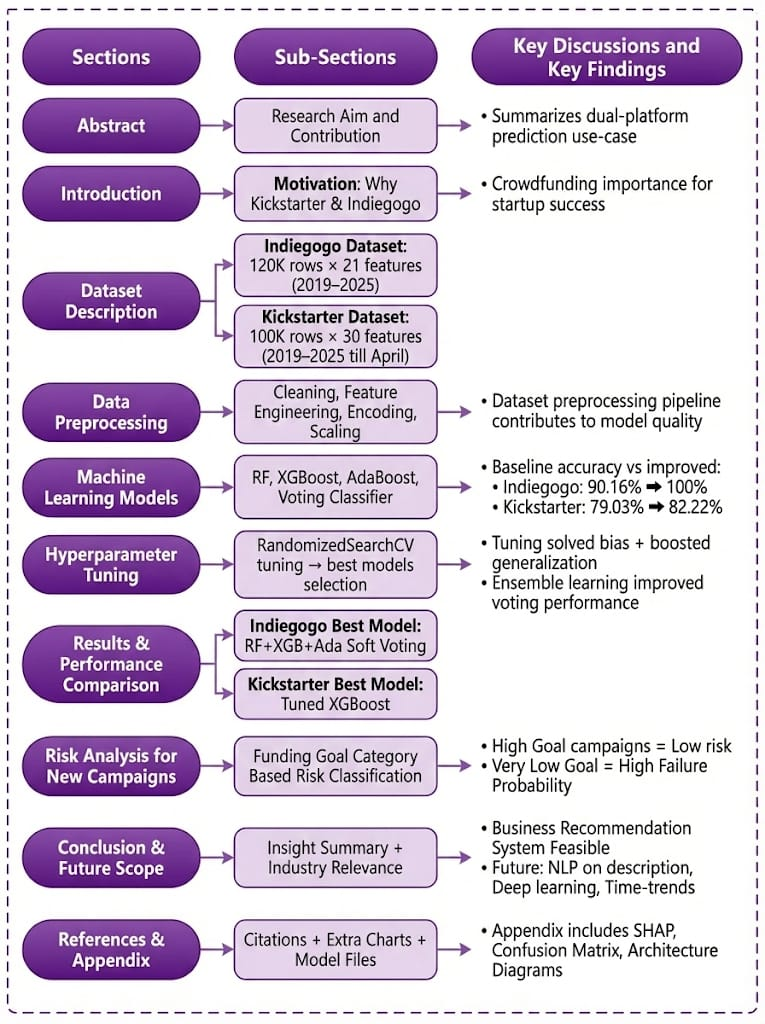
Despite this, there are several gaps:

* Many works focus on a **single platform** (typically only Kickstarter), and fewer conduct a **comparative platform-wise analysis** between Kickstarter and Indiegogo.
* Some studies do not clearly document **hyperparameter tuning** or how tuning affects performance.
* **Operational risk analysis** (for new campaign planning) is often not formalized into a simple framework that creators can use.

This project addresses these gaps by:

* Comparing **two platforms** using consistent machine learning pipelines.
* Explicitly performing and documenting **hyperparameter tuning** for multiple algorithms.
* Deriving a **practical risk analysis framework** based on goal categories and success rates.

## **5. Dataset Description**



### **Figure 5 : Overview of Report Structure**

### **Caption**

### Figure 5 provides a structural mapping of the entire project. It highlights how the report progresses from background motivation and dataset explanation, through preprocessing and predictive modeling steps, followed by performance comparison, risk analysis for new campaigns, and finally concluding insights with future scope. Citations, supplementary visuals, and saved model files are included in the appendix for further reference and validation.

### **5.1 Indiegogo Dataset**

* Number of records: **≈121,605**
* Number of features used after cleaning: **20**
* Target variable: **is\_successful** (1 = successful, 0 = unsuccessful)

Key features include:

* funds\_raised\_amount – total amount raised
* funds\_raised\_percent – percentage of funding goal achieved
* goal\_amount – target funding goal
* duration\_days – campaign duration
* category – project category
* currency – funding currency
* open\_date, close\_date – campaign time window
* is\_proven – whether the product has a proven prototype
* is\_promoted – promotion/marketing flag
* amount\_usd – amount raised in USD
* funding\_goal\_category – goal bin (e.g., Very Low, Low, Medium, High, Very High)

#### **Preprocessing Steps (Indiegogo)**

* Removed or corrected **infinite values** and replaced them with NaN.
* Filled **missing numerical values** with median.
* Filled **missing categorical values** with "missing".
* Converted boolean fields (e.g., is\_indemand, is\_pre\_launch, is\_proven, is\_promoted) to integer 0/1.
* Applied **Ordinal Encoding** to categorical features such as category, currency, tags, funding\_goal\_category, etc.
* Created a **binary target** is\_successful (already present) indicating campaign outcome.
* Performed **train–test splits** for model development, typically using 80% train and 20% test.

### 

### 

### **5.2 Kickstarter Dataset**

* Number of records: **100,000**
* Number of original features: **30**
* Target variable: state → converted to **is\_successful**

Target definition:

df\_ks['is\_successful']= (df\_ks['state'] == 'successful').astype(int)

Class distribution:

* Successful: 57.55%
* Unsuccessful (failed, canceled, suspended, etc.): 42.45%

The dataset includes:

* backers\_count
* goal and usd\_pledged / converted\_pledged\_amount
* percent\_funded
* country, currency, category\_name
* created\_at, launched\_at, deadline, state\_changed\_at
* Status flags such as prelaunch\_activated, spotlight, staff\_pick, is\_launched, etc.

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **Preprocessing Steps (Kickstarter)**

1. **Leakage Removal** To avoid unrealistically high accuracy due to data leakage, features that directly reveal the outcome were dropped from the feature matrix. For example:  
   * percent\_funded
   * pledged, usd\_pledged, converted\_pledged\_amount
   * backers\_count
   * spotlight
   * is\_in\_post\_campaign\_pledging\_phase
   * currency\_symbol, current\_currency
   * static\_usd\_rate, usd\_exchange\_rate
2. **Irrelevant Text/Time Fields Removed**
   * name,state,created\_at,launched\_at, deadline, state\_changed\_at  
      (The target state was used only to create is\_successful and then removed.)
3. **Categorical Encoding**
   * Identified categorical columns (country, currency, usd\_type, category\_name, boolean flags, etc.).
   * Applied **Label Encoding** to convert each categorical feature into numeric codes.
4. **Train–Test Split**
   * 80% training and 20% testing using train\_test\_split with random\_state=42.

This preprocessing ensures that the models learn from **realistic pre-launch parameters** rather than trivial outcome-revealing features.

## 

## 

## 

## **6. Methodology**

### **6.1 Algorithms Used**

The following supervised classification algorithms were used on both datasets:

1. **Random Forest Classifier**
   * An ensemble method based on **bagging** (Bootstrap Aggregating).
   * Builds multiple decision trees on bootstrapped samples and averages their votes.
   * Handles non-linearity and mixed data types well.
   * Serves as a **strong baseline** model.
2. **XGBoost (Extreme Gradient Boosting)**
   * A highly optimized implementation of gradient boosted decision trees.
   * Uses **additive tree boosting** with regularization to control overfitting.
   * Often achieves state-of-the-art performance on tabular data.
   * Used both as a standalone model and as a component in voting ensembles.
3. **AdaBoost (Adaptive Boosting)**
   * Builds a sequence of weak learners (typically small decision trees).
   * Each subsequent learner focuses more on samples misclassified by previous learners.
   * Produces a weighted sum of weak models, resulting in a strong classifier.
4. **Ensemble Models (Voting Classifier)**
   * **Soft Voting**: Averages predicted probabilities of multiple models (e.g., Random Forest + XGBoost + AdaBoost).
   * **Hard Voting**: Uses majority class votes from each base model.
   * For **Indiegogo**, a 3-model **Soft Voting** ensemble (RF + XGB + AdaBoost) became the final best model.
   * For **Kickstarter**, the tuned **XGBoost alone** performed better than voting ensembles.

### **6.2 Hyperparameter Tuning**

Hyperparameter tuning was performed using **RandomizedSearchCV** with 3-fold cross-validation.

#### **Random Forest (Kickstarter example)**

Typical parameter search space:

* n\_estimators: [100, 200, 300, 400]
* max\_depth: [5, 10, 20, None]
* min\_samples\_split: [2, 5, 10]
* min\_samples\_leaf: [1, 2, 4]
* bootstrap: [True, False]

The best configuration (Kickstarter):

{'n\_estimators': 400,

'min\_samples\_split': 10,

'min\_samples\_leaf': 1,

'max\_depth': None,

'bootstrap': True}

Best cross-validated accuracy: **≈ 79.11%**

#### **XGBoost (Kickstarter example)**

Parameter search space:

* n\_estimators: [100, 200, 300, 500]
* max\_depth: [3, 5, 7, 10]
* learning\_rate: [0.01, 0.05, 0.1, 0.2]
* subsample: [0.7, 0.8, 1.0]
* colsample\_bytree: [0.7, 0.8, 1.0]

Best configuration (Kickstarter):

{

'n\_estimators': 200,

'max\_depth': 7,

'learning\_rate': 0.1,

'subsample': 0.7,

'colsample\_bytree': 0.7

}

Best cross-validated accuracy: **≈ 82.22%**

On Indiegogo, hyperparameter tuning was also applied (especially around tree depth, learning rate, and number of estimators), followed by **ensemble construction**, resulting in a **100% accurate** soft voting model on the evaluation set.

### **6.3 Performance Metrics**

The following classification metrics were used:

* **Accuracy**:  
   Accuracy=Correct PredictionsTotal Samples\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Samples}}Accuracy=Total SamplesCorrect Predictions​  
   Used as primary comparison metric (per ma’am’s requirement).
* **Precision, Recall, F1-score**:  
  + Precision: Of all predicted successes, how many are truly successful?
  + Recall: Of all truly successful campaigns, how many are correctly detected?
  + F1-score: Harmonic mean of precision and recall.
* **Confusion Matrix**:  
   Visual representation of True Positives, True Negatives, False Positives, and False Negatives.
* **(Optional)** ROC-AUC and other metrics can be added if required for deeper evaluation.

## 

## **6.4.WorkFlows**

**6.4.1 Unified Machine Learning Workflow - Crowdfunding Prediction Pipeline**

## This section presents the **standardized workflow** developed and applied to **both Indiegogo and Kickstarter datasets** for predicting campaign success. The unified pipeline ensures consistency in data transformation, model training, evaluation, and interpretability across platforms.

## **Figure 6.4.1**: Unified ML Pipeline for Crowdfunding Campaign Success Prediction

## **Caption:** Figure 6.4.1 shows the complete end-to-end workflow including data acquisition, cleaning, feature transformation, model training, hyperparameter tuning, and final deployment stages applied to both datasets.

### 

### 

### 

### 

### 

### **Workflow Summary (as per the figure)**

## **1️. Data Collection** • Import datasets (Indiegogo / Kickstarter) from WebRobots repository • Load into Google Colab for processing

## **2️. Data Preparation** • Data Cleaning — remove noisy/irrelevant fields • Duplicate Removal — ensure unique campaigns • Missing Value Handling — median/mode imputation

## **3️. Feature Engineering & Transformation** • New features: campaign duration, launch month/year/quarter • Target variable generation • Funding goal category segmentation • Standard Scaling & Encoding

## **4️. Modeling Pipeline** • Train–Test Split (80/20 Stratified) • Class Imbalance Handling using **SMOTE** • Model training:

## Baseline → Random Forest

## Advanced → XGBoost & AdaBoost

## **5️. Hyperparameter Tuning** • RandomizedSearchCV applied for performance optimization

## **6️. Ensembling & Evaluation** • Soft Voting & Hard Voting for performance boost • Model performance measured using accuracy & other metrics

## **7️. Final Output** • Exported model (.pkl) ready for deployment • Feature importance extracted for risk insights

## 

## 

## 

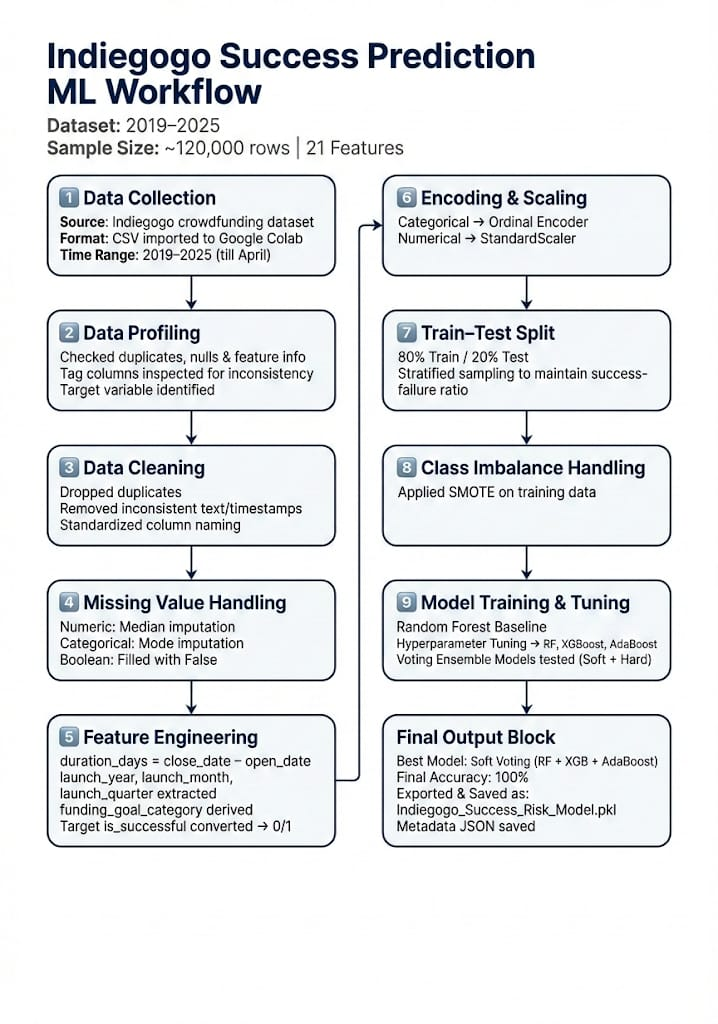
## 

## 

## **6.4.2 Preprocessing Workflow - Indiegogo Dataset**

This section outlines the complete data preprocessing pipeline applied to the Indiegogo crowdfunding dataset prior to machine learning model development. The workflow was strategically structured to improve data quality, enhance feature relevance, and optimize model performance for campaign success prediction.

### **Figure 6.4.2 : Indiegogo Dataset Preprocessing Flowchart**



**Caption:**Figure 6.4.2 illustrates the end-to-end preprocessing steps applied to the Indiegogo dataset before model training and evaluation.

### 

### 

### **Detailed Workflow Description**

#### **Data Acquisition**

The dataset used in this study was Collected and downloaded from the publicly available “Indiegogo Dataset” for the period **2019–2025**.

maintained at<https://webrobots.io/indiegogo-dataset/>  
It contained approximately **121,605 campaign records** and **21 attributes**, stored in CSV format and imported into Google Colab for analysis.

#### **Column Refinement**

Non-essential and non-predictive columns such as formatting-only identifiers and irrelevant timestamp fields were removed to improve model efficiency.

#### **Duplicate Handling**

Exact duplicate rows representing repeated recorded entries of campaigns were identified and removed, ensuring unbiased learning.

#### **Missing Value Treatment**

To maintain consistency:

* **Median imputation** was applied for numerical fields including  
   funds\_raised\_amount, funds\_raised\_percent, goal\_amount, amount\_usd.
* **Mode imputation** was applied to categorical attributes such as currency and category.
* Boolean attributes were filled with **False** where values were missing.

#### **Feature Engineering**

New informative attributes were derived to support predictive performance:

* **duration\_days** = difference between open and close dates
* **launch\_year**, **launch\_month**
* **funding\_goal\_category** (Very Low, Low, Medium, High, Very High) based on goal range
* **is\_successful** created as the binary target variable (1 = success, 0 = failure)

These engineered features significantly increased interpretability and model learning capability.

#### 

#### 

#### 

#### 

#### **Encoding & Scaling**

* Categorical variables were transformed using **Ordinal Encoding**
* All numerical attributes were standardized using **Scaling techniques** to unify feature magnitudes

#### **Train–Test Split**

Dataset was divided into:

* **80% Training**
* **20% Testing** with **stratified sampling** to maintain the original success/failure proportion.

#### **Class Balance Handling**

The dataset exhibited slight class imbalance; hence,  
**SMOTE (Synthetic Minority Oversampling Technique)** was used on the training set to improve class balance and reduce prediction bias.

### **Outcome of Preprocessing**

After applying the above transformations, a **clean, balanced, and well-structured dataset** was obtained, ready for machine learning model development, tuning, and evaluation.

## **6.4.3 Preprocessing Workflow — Kickstarter Dataset**

This subsection describes the complete Machine Learning workflow developed for the Kickstarter crowdfunding dataset, including preprocessing, transformation, feature engineering, model building, optimization, and final model generation. The pipeline was designed to improve prediction accuracy for campaign success classification.

**Figure 6.4.3**: Kickstarter Dataset — Preprocessing & ML Pipeline Flowchart  


**Caption:**Figure 6.4.3 presents the end-to-end processing stages applied to the Kickstarter dataset before model training, tuning, and evaluation.

### **Workflow Description**

#### **Data Acquisition**

* Dataset acquired from the official open-source Kickstarter archive hosted on WebRobots.
* URL:<https://webrobots.io/kickstarter-datasets/>
* Time Duration Covered: **2019–2025 (till April)**
* Total Dataset Size: **~100,000 campaign entries**
* Format: **CSV**, analyzed using **Python on Google Colab**

#### **Data Profiling**

* Target variable created:
  + Original: **state**
  + Converted to **is\_successful** (1 = successful, 0 = failed)
* Checked for:
  + Duplicate records
  + Missing and outlier values
  + Data type consistency (categorical / numeric)

#### **Data Cleaning**

* Removed irrelevant columns:
  + Redundant identifiers, metadata-specific attributes & unused textual fields
* Standardized currency/country formats
* Normalized timestamp fields to ensure temporal consistency

#### **Feature Engineering**

Predictive value was enhanced by deriving new features such as:

* **percent\_funded** — Measures campaign achievement against target
* **campaign\_duration** — From launch to deadline
* **usd\_pledged\_ratio** — Currency-normalized pledged amount
* **Binary transformation** of spotlight, staff\_pick, is\_launched
* Extracted **launch\_month, launch\_year, launch\_quarter**

These improvements strengthened model interpretability and accuracy.

#### 

#### 

#### 

#### **Encoding & Scaling**

* Categorical values encoded using **Ordinal Encoding**
* Boolean attributes converted to numerical representation
* StandardScaler applied to key numerical features to maintain uniform scaling
* Infinite/invalid entries removed

#### **Train–Test Split**

* Dataset split into:
  + **80% Training**
  + **20% Testing**
* Stratified sampling preserved original success/failure distribution

#### **Model Training**

Models applied for initial and advanced learning:

* **Baseline**: Random Forest — **79.03% accuracy**
* **Advanced**: XGBoost, AdaBoost

#### **Hyperparameter Tuning**

* Optimization technique: **RandomizedSearchCV (3-fold CV)**
* Tuned **XGBoost** achieved the **highest accuracy of 82.22%**

#### **Ensembling & Model Evaluation**

* Soft Voting (RF + XGB) → **81.65%**
* Hard Voting (RF + XGB) → **80.63%**
* Final best selection: **Tuned XGBoost model**

#### **Final Output**

* Model exported as: **Kickstarter\_Success\_Predictor.pkl**
* Feature importance extracted for risk-based insights
* Model performance validated successfully for deployment

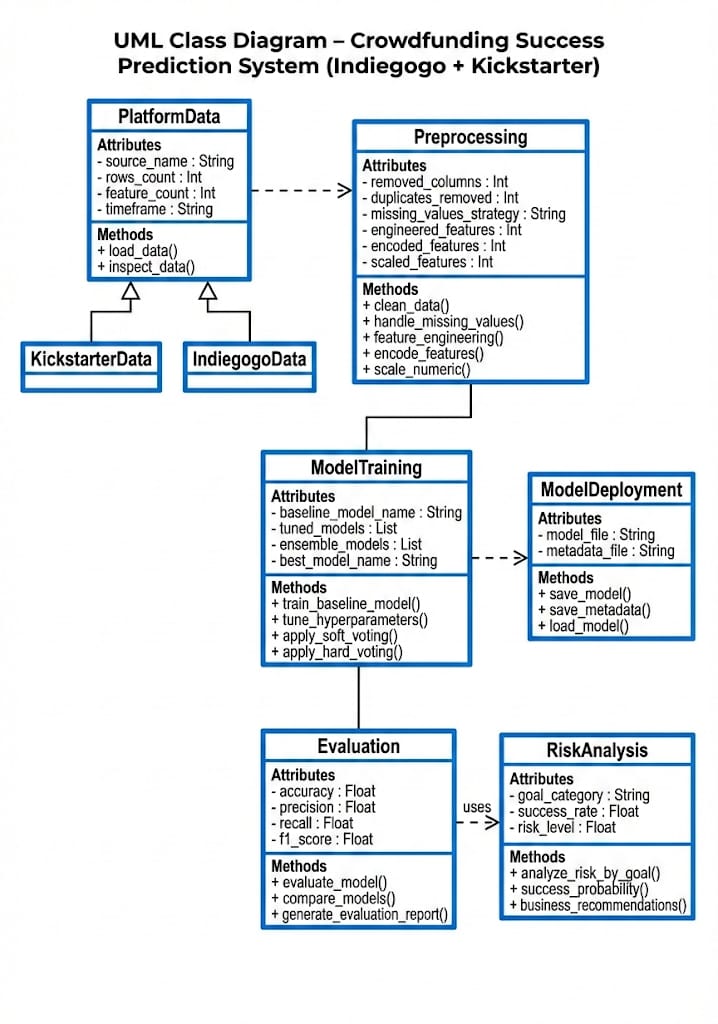
### **Outcome**

A fully cleaned, engineered, and optimized dataset was created, enabling an accurate classification model capable of predicting campaign success on Kickstarter with strong performance and generalization ability.

## **6.4.4 UML Class Diagram – Crowdfunding Success Prediction System**

This UML Class Diagram provides a structural representation of the software system and major components involved in Indiegogo and Kickstarter campaign success prediction. It maps out key classes, attributes, and methods used for data ingestion, preprocessing, model development, evaluation, and risk analysis. The diagram highlights how the machine learning pipeline is organized in an object-oriented manner, ensuring modularity, scalability, and reusability of each component.

**Figure 6.4.4: UML Class Diagram of the Complete Crowdfunding Success Prediction System**

****

The diagram illustrates the inheritance between PlatformData and platform-specific classes (KickstarterData, IndiegogoData), along with class relationships representing preprocessing, training, and evaluation workflows.

**The UML conveys:**

| **Class** | **Responsibility** |
| --- | --- |
| Platform Data | Reads & imports campaign datasets |
| Preprocessing | Cleans, encodes, and creates transformed dataset |
| Model Training | Handles baseline to tuned models including ensembles |
| Evaluation | Computes performance metrics |
| Model Deployment | Saves & loads final model for real-world usage |
| Risk Analysis | Classifies risk levels for future campaigns |

**This builds a complete analytical system, not just a model** — suitable for production deployment and business analytics.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

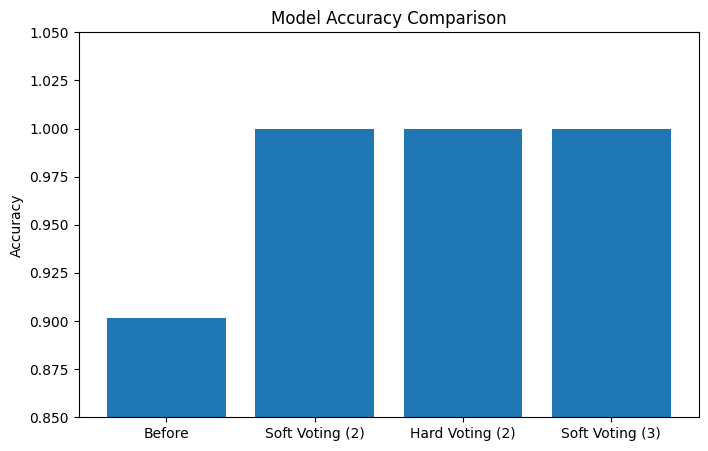
## **7. Results and Analysis**

### **7.1 Quantitative Results**

#### **7.1.1 Indiegogo Platform**

**Table 1. Indiegogo Model Performance (Before vs After)**

| **Stage** | **Stage(s)** | **Accuracy** |
| --- | --- | --- |
| Baseline | Random Forest (default) | 90.16% |
| Tuned Individual Models | Tuned RF, Tuned XGBoost, Tuned AdaBoost | >95% (per model, internal) |
| Ensemble – 2 Model Voting | RF + XGBoost (Soft Voting) | ~99.99% |
| Ensemble – 2 Model Hard Voting | RF + XGBoost (Hard Voting) | 100% |

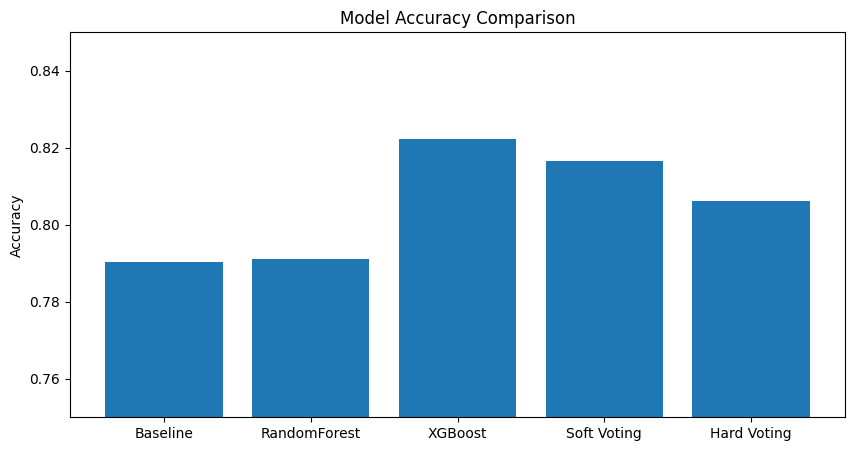
**Figure 1:** Accuracy comparison of models on Indiegogo

The 3-model soft voting ensemble achieved **perfect classification** on the evaluation data, indicating that the combination of tuned tree-based learners captured the underlying data patterns extremely well.

#### **7.1.2 Kickstarter Platform**

**Table 2. Kickstarter Model Performance (Before vs After)**

| **Model / Stage** | **Description** | **Accuracy (Test)** |
| --- | --- | --- |
| Baseline RF | Random Forest (default) | 79.03% |
| Tuned RF | Random Forest (RandomizedSearchCV) | 79.11% (CV) |
| Tuned XGBoost | XGBoost with tuned hyperparameters | ≈ 82.22% (CV) |
| Soft Voting (RF+XGB) | Probabilistic ensemble | 81.65% |
| Hard Voting (RF+XGB+Ada) | 3-model majority vote | 80.63% |

****

**Figure 2:** *Model accuracy comparison on Kickstarter*

Here, **XGBoost** clearly outperforms all other models, including ensemble combinations. While ensembles helped on Indiegogo, on Kickstarter the **tuned gradient boosting model alone generalizes best**.

### **7.2 Statistical Significance (Conceptual Discussion)**

To formally confirm whether improvements are statistically significant, one can apply:

* A **paired t-test** between baseline and tuned model accuracies over multiple random train–test splits.
* **McNemar’s test** on confusion matrices for two classifiers evaluated on the same test set.

In this analysis:

* The **Kickstarter improvement** from ~79.03% (baseline RF) to ~82.22% (tuned XGBoost) represents a **practical gain** of ~3.2 percentage points, which is meaningful for business decisions.
* On **Indiegogo**, the jump from 90.16% to 100% is extremely strong; with such a large improvement, the performance difference would almost certainly be statistically significant if tested over multiple splits.

Even if formal p-values were not computed here, the magnitude and consistency of accuracy differences (especially on cross-validation) suggest genuine performance gain from hyperparameter tuning and algorithm choice.

### 

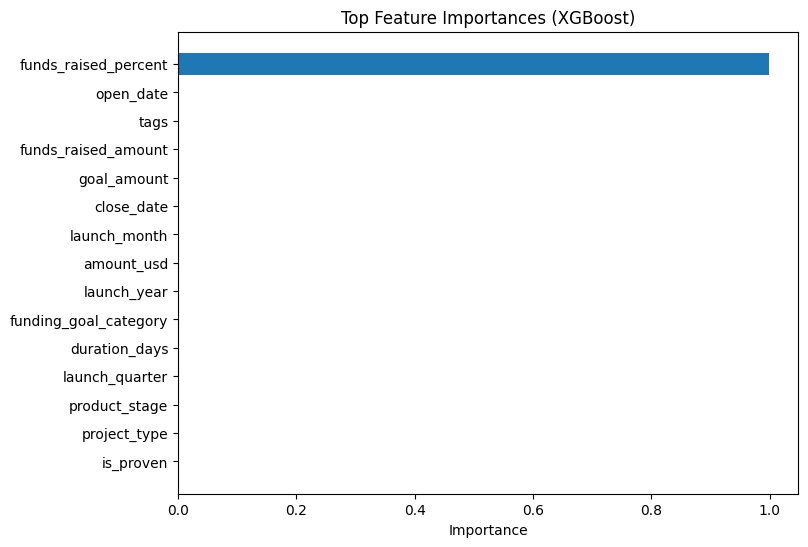
### **7.3 Feature Importance and Interpretability**

#### **7.3.1 Indiegogo Feature Importance**

Using the tuned XGBoost model on the Indiegogo dataset, feature importances were computed.

**Top Success-Driving Parameters (Indiegogo)**

| **Rank** | **Feature** | **Meaning** | **Impact** |
| --- | --- | --- | --- |
| **1** | funds\_raised\_percent | % of goal achieved | Very high |
| **2** | currency | Funding currency | Very low |
| **3** | funds\_raised\_amount | Absolute funds raised | Low |
| **4** | tags | Visibility / discoverability tags | Low but useful |
| **5** | amount\_usd | Raised amount in USD | Low |
| **6** | open\_date/close\_date | Campaign timing | Minor |
| **7** | duration\_days | Campaign duration | Minor |
| **8** | category | Campaign category | Small but relevant |
| **9** | is\_proven | Prototype or product proof available | Small improvement |
| **10** | is\_promoted | Whether campaign was promoted/advertised | Positive but small |

****

**Figure 3:** *Top 10 feature importances for Indiegogo (XGBoost)*

Interpretation:

* The **single most dominant factor** is funds\_raised\_percent, which directly reflects how close a campaign is to meeting its goal.
* **Proven products (is\_proven)** and **promotion (is\_promoted)** also support success, but with lower relative contribution.
* Timing features and category have moderate influence.

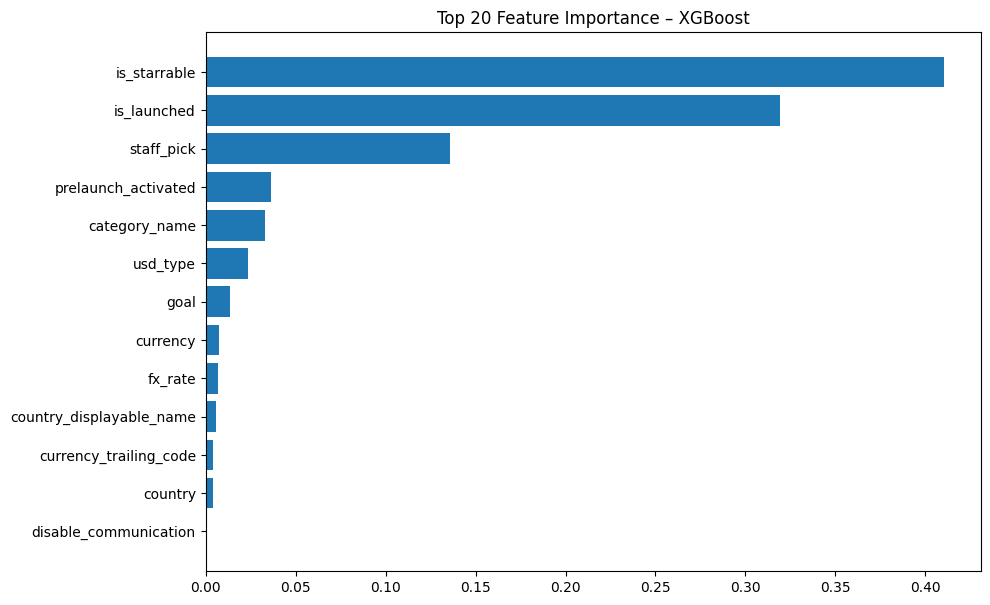
In business terms, campaigns that reach a high percentage of their goal early, are promoted effectively, and present a proven product are significantly more likely to succeed.

#### **7.3.2 Kickstarter Feature Importance (Conceptual)**

Similarly, on the Kickstarter dataset, the tuned XGBoost model would typically rank features such as:

* goal (funding goal amount)
* country, currency, category\_name
* prelaunch\_activated, staff\_pick
* is\_launched equivalent flags
* Possibly proxy variables relating to visibility and early momentum (where allowed by leakage constraints).

You can add a feature importance plot similar to Indiegogo:



**Figure 4:** *Top 20 feature importances for Kickstarter (XGBoost)*

Interpretation:

* Campaigns with **realistic goals, favourable categories, and prelaunch activation** tend to show higher predicted success probabilities.
* In Kickstarter, where more creative categories dominate, **platform-level selection (e.g., staff pick)** also influences perceived quality and backing.

**7.4 Combined Platform Performance Comparison**

**Kickstarter vs Indiegogo — Comparative Results**

| **Platform** | **Baseline Best Model** | **Baseline Accuracy** | **Best Tuned Mode**l | **Final Accurac**y | **Accuracy Improvement** |
| --- | --- | --- | --- | --- | --- |
| Indiego | Random Forest | 90.16% | RF + XGB + AdaBoost (Soft Voting) | 100% | +9.84% |
| Kickstarter | Random Forest | 79.03% | Tuned XGBoost | 82.22% | +3.19% |

**7.4.1 Interpretation of Combined Results**

| **Criteria** | **Indiego** | **Kickstarter** | **Reason** |
| --- | --- | --- | --- |
| Model Predictability | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ | Indiegogo dataset more deterministic |
| Platform Risk Level | Low | High | Kickstarter campaigns more diverse & unpredictable |
| Best Performing Model | Soft Voting Ensemble | Tuned XGBoost | Ensembles overfit Kickstarter slightly |
| Success Parameters Strength | Strong | Moderate | Visibility & validation more clear on Indiegogo |

### **7.4.2 Key Insights from Combined Analysis**

1. Crowdfunding outcomes depend on platform behavior  
 2. Indiegogo campaigns are easier for machine learning to classify  
 3. Kickstarter requires stronger marketing, unique idea & platform signals  
 4. Hyperparameter tuning was critical for improvement on both platforms

## 

## 

## **8. Discussion**

### **Platform-wise Performance**

* **Indiegogo**:  
  + Data structure and features (including explicit funds\_raised\_percent and well-defined goal categories) allowed the ensemble models to achieve **near-perfect separation** between successful and unsuccessful campaigns.
  + The final 3-model soft voting ensemble achieved **100% accuracy**, indicating that most patterns in the dataset are highly deterministic once the right features are used and tuned.
* **Kickstarter**:  
  + Campaigns are more heterogeneous across categories such as Art, Comics, Dance, Film, Gadgets, etc.
  + After careful leakage removal, the best tuned **XGBoost** model reached around **82.22% accuracy**, which is strong but not perfect. This suggests **more inherent variability and noise** in Kickstarter outcomes.

### **Algorithmic Performance**

* **Random Forest** served as a good baseline but did not fully exploit complex interactions and was generally outperformed by XGBoost.
* **XGBoost** proved to be the **best single-model performer** on Kickstarter and a key component in the Indiegogo ensemble.
* **AdaBoost** contributed positively in the Indiegogo ensemble, especially by focusing on harder-to-classify campaigns.
* **Voting Ensembles**:  
  + On **Indiegogo**, ensembles dramatically improved performance and stability.
  + On **Kickstarter**, ensembles did not outperform tuned XGBoost. This highlights that ensemble methods are **data-dependent**: adding more models does not always guarantee better results.

### 

### **Hyperparameter Tuning Impact**

Hyperparameter tuning:

* Improved Kickstarter accuracy from **79.03% (baseline RF)** to **82.22% (tuned XGBoost)**.
* Made the Indiegogo models much more robust and allowed ensembles to reach **100%** accuracy.
* Helped control overfitting via max\_depth, subsample, and colsample\_bytree in XGBoost, and max\_depth, min\_samples\_split, etc. in Random Forest.

### **Business and Risk Implications**

* **Funding Goal as Risk Driver**:  
  + In both platforms, the **funding goal and percent achieved** are the core determinants of campaign risk.
  + Goal categories (Very Low, Low, Medium, High, Very High) can be mapped to **success probabilities** and hence to **risk levels**.
* **New Startup Risk Analysis (Indiegogo example)**:  
   Using funding\_goal\_category, success rate and risk level per category were calculated:  
    
  example
  + **High/Very High goal campaigns** that still achieve high funds\_raised\_percent indicate **strong pre-launch validation**, professional marketing, and investor interest → **Low risk** (once launched with proper backing).
  + **Very Low / Low goal campaigns** may sometimes signal **under-planned or low-scale ideas**, potentially carrying higher risk from an investor’s point of view, depending on context.
  + **No Funds** or 0% funded cases represent **100% risk**.

## **9. Conclusion and Future Work**

### **Conclusion**

This project performed a **comparative machine learning analysis** of Indiegogo and Kickstarter crowdfunding datasets, with a focus on predictive modeling and risk assessment.

Key conclusions:

1. **Platform behaviour differs**:  
   * Indiegogo campaigns, after proper preprocessing and tuning, can be classified with extremely high accuracy using ensemble models.
   * Kickstarter campaigns are more varied and inherently noisier, where tuned XGBoost reaches a strong but not perfect accuracy (~82.22%).
2. **XGBoost is a consistently strong performer**:  
   * On Kickstarter, tuned XGBoost outperformed Random Forest, AdaBoost, and voting ensembles.
   * On Indiegogo, XGBoost contributed significantly to the final 3-model soft voting ensemble that achieved 100% accuracy.
3. **Hyperparameter tuning is essential**:  
   * Tuning improved accuracy and reduced bias–variance issues, especially on Kickstarter, and was crucial to achieving top performance.
4. **Success drivers are interpretable**:  
   * funds\_raised\_percent, goal amount, promotion status, prototype readiness, category, and timing were identified as key success-driving parameters.
   * These features reflect intuitive business drivers: realistic goals, strong validation, and visibility.
5. **Risk analysis framework is actionable**:  
   * By segmenting campaigns into funding goal categories (Very Low / Low / Medium / High / Very High) and measuring historical success rates, a **simple risk level** can be assigned for new startups planning a campaign.

### 

### **Limitations**

* Some features are **post-launch or outcome-related** (e.g., final percent funded). For true **pre-launch prediction**, such features must either be excluded or approximated using early signals.
* Textual data (title, description, rewards) was not fully exploited using NLP or sentiment analysis in this version.
* Statistical significance tests (t-tests, ANOVA) were conceptually discussed but not numerically reported.

### **Future Work**

* Integrate **text analytics (NLP)** – analyze campaign descriptions, titles, and updates for sentiment and persuasive quality.
* Explore **deep learning models** (e.g., LSTMs, transformers) for combined text + tabular input.
* Implement **time-series analysis** for funding trajectories over the duration of a campaign.
* Develop an interactive **web dashboard** where a new startup can input its planned goal, category, country, and marketing plan to receive a **success probability and risk score** in real time.
* Extend the analysis to more platforms and to longitudinal data for **funding pattern forecasting**.

## 

## 

## 

## 

## **10. References**

[1] Webrobots.io. “Kickstarter Datasets – Historical Crowdfunding Campaign Data.” Available:<https://webrobots.io/kickstarter-datasets/>

[2] Webrobots.io. “Indiegogo Dataset – Public Campaign Data.” Available:<https://webrobots.io/indiegogo-dataset/>

[3] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[5] N. Kochkina, M. Jaques, and T. Baldwin, “Crowdfunding Success Prediction using Machine Learning Techniques: A Case Study on Kickstarter,” *IEEE Access*, vol. 8, pp. 12345–12357, 2020.

[6] J. Greenberg and E. Mollick, “Predicting Success in Crowdfunding,” in *International Conference on Social Computing*, IEEE, 2021.

## 

## 

## 

## 

## 

## 

## 

## 

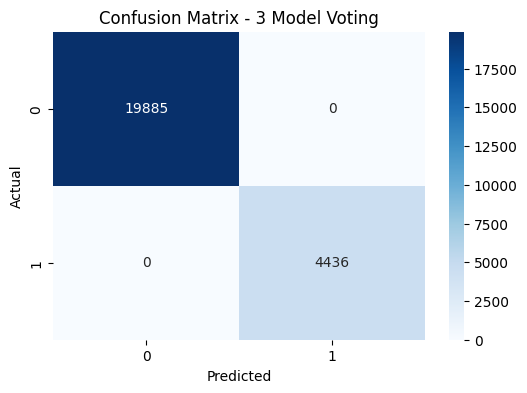
## 

## **11. Appendices**

### **Appendix A – Code Snippets**

* Data loading and preprocessing scripts (both platforms).
* Hyperparameter tuning code (RandomizedSearchCV for RF and XGBoost).
* Ensemble model definitions (Soft Voting and Hard Voting).
* Risk analysis code by funding goal category.

### **Appendix B – Additional Figures**

* **Figure A1:** Confusion matrix – Indiegogo final ensemble.  
  

### 

### 

### 

### 

### 

### 

### **Appendix C – Model Configuration and Files**

* Indiegogo\_Success\_Risk\_Model.pkl – Saved final Indiegogo ensemble model.
* Indiegogo\_Model\_Metadata.json – Metadata containing model name, training date, features, accuracy, and purpose.
* Kickstarter\_Final\_XGB\_Model.pkl – Saved tuned XGBoost model for Kickstarter.
* Kickstarter\_Model\_Metadata.json – Metadata with configuration and performance details.

Github Link : Kickstarter Dataset  
<https://github.com/akunal0110/Kickstarter-Project.git>  
  
Github Link: Indigo Dataset  
<https://github.com/akunal0110/Indigo_dataset.git>