Predicting Kickstarter Projects Success.

A Data Science & Machine Learning Case Study.

Ali Alsudani

23.01.2026

Kickstarter Overview:

What is Kickstarter?

Online crowdfunding platform (launched in 2009).

Supports creative projects:

- Art, Film, Games, Design, Technology

How funding works

Backers pledge money to projects they like

All-or-nothing model:

- Goal reached → project funded
- Goal not reached → no money collected

Key idea:

Success is determined at launch deadline

Why Kickstarter Project Success Matters (Business Perspective)

For Creators

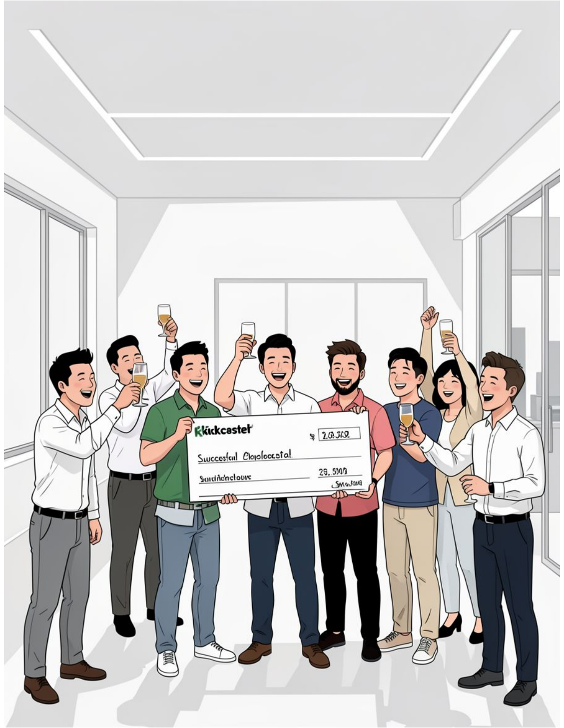
- Funding is received only if the project succeeds
- Helps creators set realistic goals and strategies

For Backers

- Backers receive rewards only if projects succeed

For Platform

- Kickstarter earns revenue only from funded projects
- Higher success rates improve platform credibility, growth and trust.



Project definition:

Problem

- Predict whether a Kickstarter project will succeed

Target:

- Successful (1) vs Failed (0)

Timing:

- Prediction made before launch

Constraint:

- Using only pre-lunch information.

Why This Is a Data Science Problem?

- **Kickstarter is well-suited for predictive modeling because**

  Each project has a clear outcome: Success or Failure

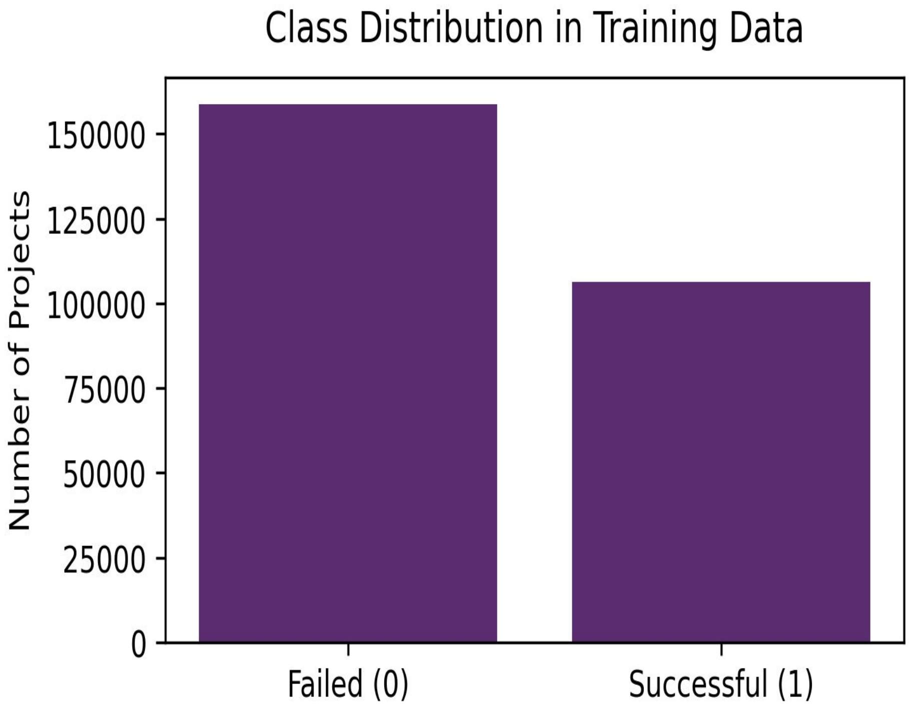
- **Many features are known before launch:**
- Large historical dataset enables learning patterns

- **Key question:**
  Can we predict project success using only information available at launch time??

Data Overview:

- **~330,000 Kickstarter projects**
- **Features available at launch:**
- **Funding goal & duration**
- **Category & country**
- **Launch timing**
- **Project name features**

➢ **Class distribution::**
- 58.5% Failed (0)
- 41.5% Successful (1)



Data Preparation Summary.

- Raw dataset: 374,853 projects
- After cleaning & filtering: 331,368 projects
- Target defined as:
  Successful (1) , Failed (0)
- Non-final outcomes removed (e.g., Live, Suspended)

Train–Test Split Strategy.

- **Time-based split to simulate real-world prediction**
  Training data:

➔ **Training data:**
 2009-04 → 2016-06
 265,094 projects

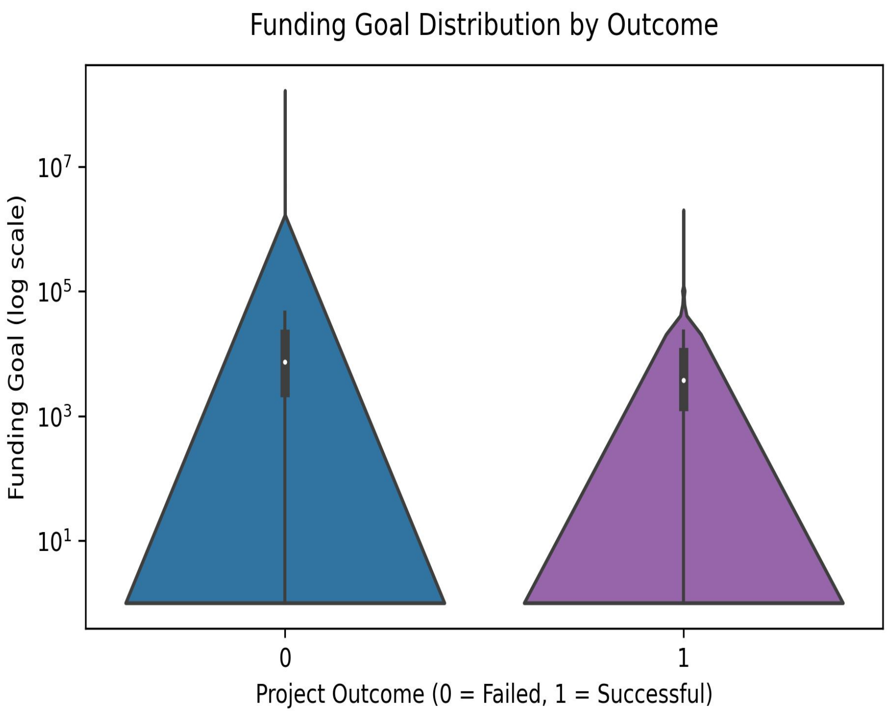
➔ **Test data:**
2016-06 → 2017-12
66,274 projects

Exploratory Data Analysis (EDA).

Funding Goal Distribution by Outcome (log scale)

Funding goals are highly skewed

- failed project tend to request higher and more dispersed funding goals.
- while successful projects cluster around more moderate targets.



Evaluation Metric.

Dataset is class-imbalanced (more failed than successful projects)

❖ Evaluation metric: ROC–AUC

We use ROC–AUC instead of accuracy or F1:

➔ Handles class imbalance
➔ Measures ranking quality
➔ Threshold independence

❖ This allows stakeholders to choose decision thresholds later based on business risk.

Baseline Model Performance.

★ Logistic Regression baseline.

Uses only basic pre-launch features

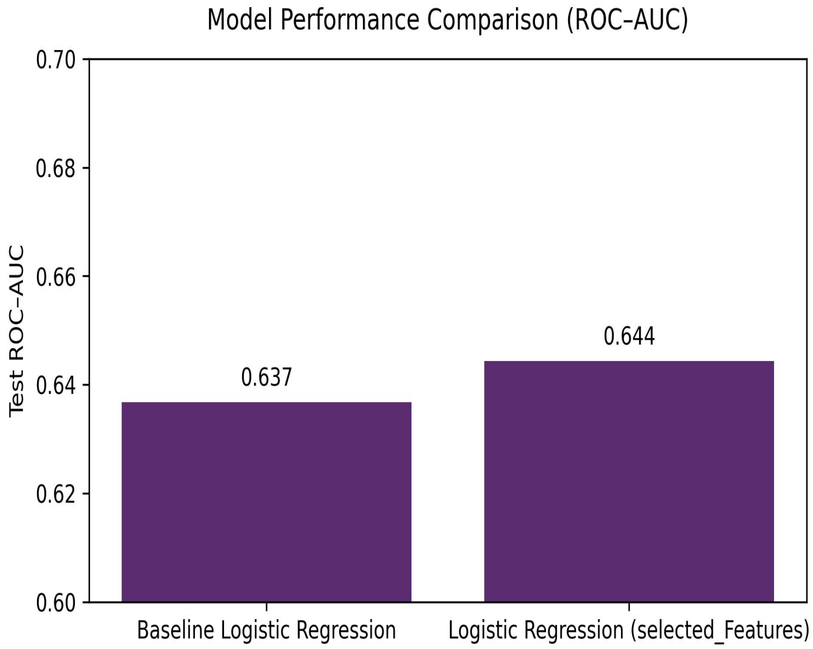
➔ (funding goal, duration, category, country, launch timing)

Key result:

➢ Test ROC–AUC ≈ 0.63

Improved Linear Model (Feature Engineering):

- Added engineered features that capture funding realism and launch context.
- Focused on a small set of high-impact features.
- Same Logistic Regression model for fair comparison.
- Evaluation remains ROC–AUC on test data.



Conclusion:

Feature engineering improves the model’s ability to distinguish successful projects before launch.

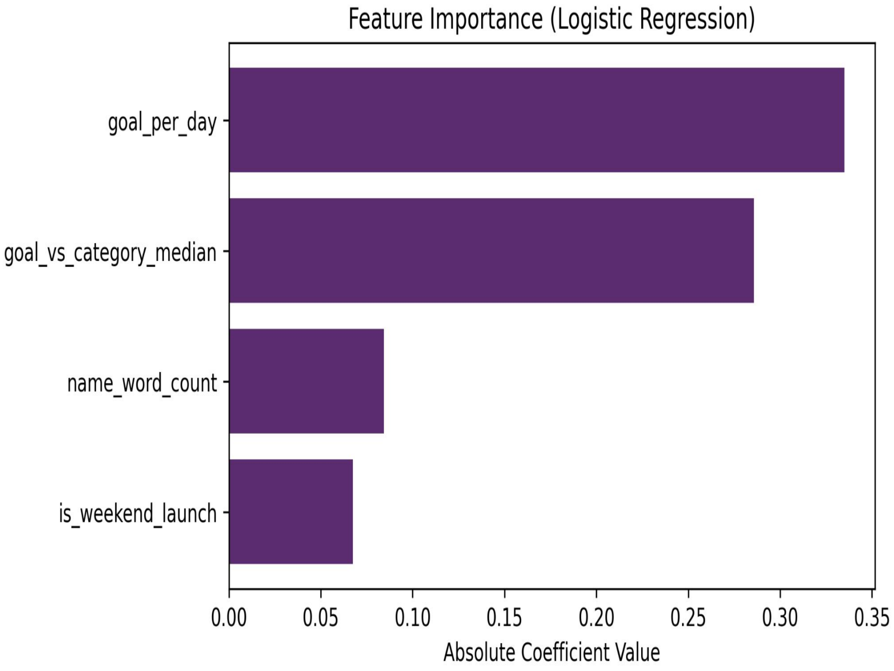
Why Performance Improved?

The plot shows goal-related features dominate model decisions

Selected features include

(goal per day, goal vs category median, name word count, and launch timing).

- Normalizing goals by context (per day, per category) improves separation.
- Text features add minor signal; timing features have limited impact



Advanced Model (XGBoost)

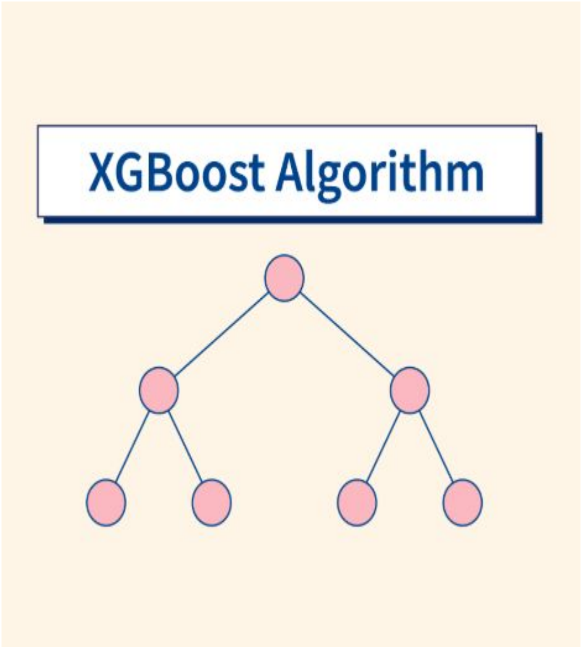
Why we move beyond logistic regression?

Why XGBoost?

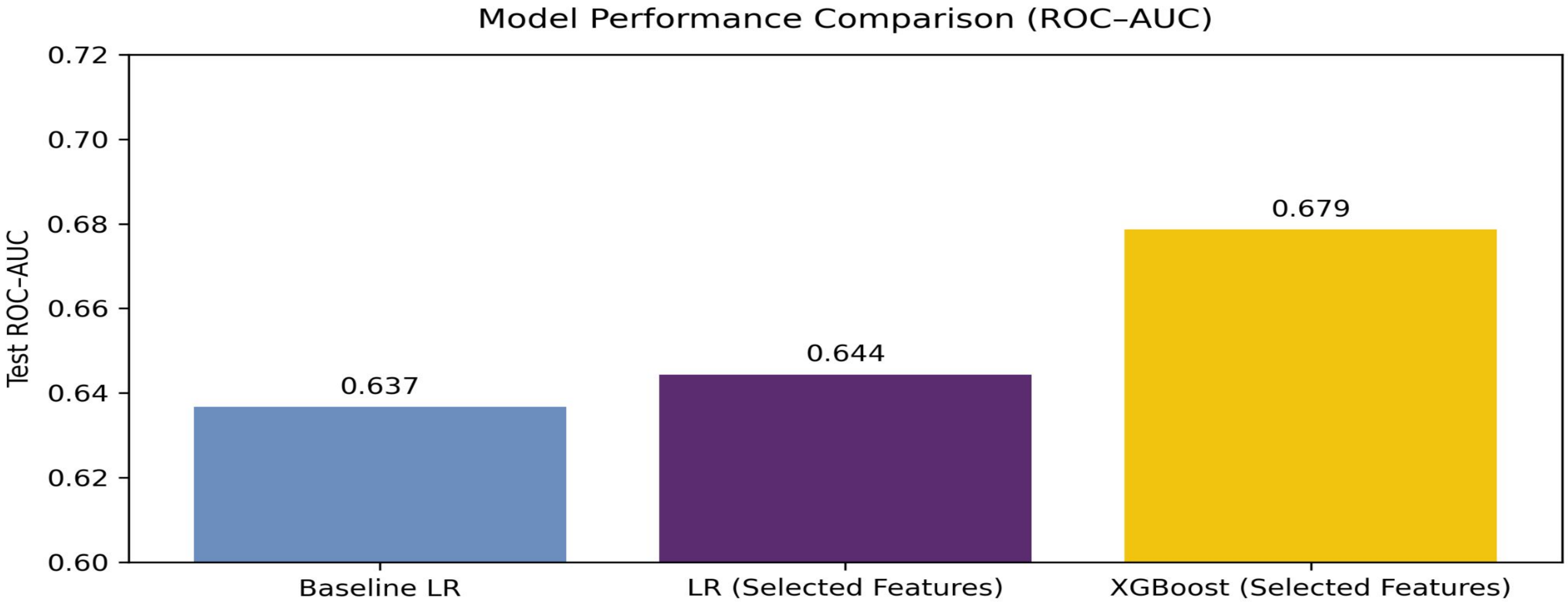
- Tree-based ensemble model.
- Automatically captures non-linear effects and feature interactions.
- Strong performance on structured/tabular data.

WE IMPLEMENT:

- Same lunch features.
- Same train/ test split.
- Same evaluation metrics (ROC-AUC).



XGBoost Performance Result:



Non-linear models extract more signal from engineered features.

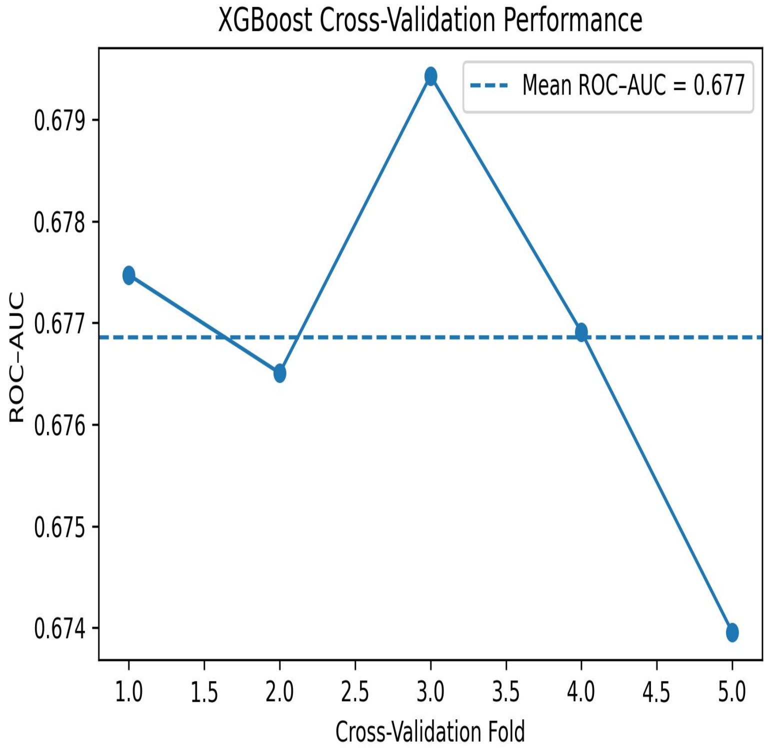
- **The chart shows clear significant result 0.67 compared to baseline model 0.63.**

Model Performance (Cross-Validation)

If the data changes slightly, does the model still perform similarly?

- Mean ROC–AUC ≈ 0.677
- Very low variance across folds (std ≈ 0.002)

- Performance is consistent across folds
- Improvement is robust, not due to chance
- Model generalizes well to unseen data



Conclusion:

- Kickstarter success can be predicted before launch.
- Feature engineering provides substantial gains.
- XGBoost delivers robust, stable improvement.
- Results are validated via cross-validation.

Data-driven insights can meaningfully improve crowdfunding outcomes.

Limitations and Future Works:

Limitations:

- Text features are simplistic.
- Limited hyperparameter tuning.
- No creator history included.

Future Work:

- NLP on project descriptions.
- Creator-level features.
- LightGBM / CatBoost comparison.
- Threshold optimization for decision support.


neue fische
SPICED

Thank You

Questions & Discussion

Data-driven insights to improve project success before launch

Ali Alsudani