Team Kickstarter

Model Definition and Initial Results

Alfonso Alday, Allison Bass, Dylan Juarez, Prakash Rao, Franklin Tan

BANA 5160 | 19-JUL-2023

Action Plan

Our team intends to use an advanced classifier to identify key factors contributing to Kickstarter campaign failures. Specifically, after preprocessing, cleaning, and splitting the data into training, validation, and test sets, we aim to try out and iteratively compare different types of machine learning algorithms.

We used logistic regression to assess which model is best suited to learn the patterns and relationships between our chosen predictors[1] and binary target variable[2].

Our main goal is to gain clear and interpretable insights about the drivers of campaign failure for Kickstarter's leadership, therefore, we will focus on explainability. Nonetheless, we will look at standard classification metrics related to performance like accuracy, precision, F1 score, and AUC-ROC to ensure the quality of our predictive model.

Scalability and implementation are not a concern for us because our dataset has 378,661 historical records and we have a robust available infrastructure of compute resources, softwares, and machine learning expertise at our disposal.

Completed Analysis

Thus far, we have performed exploratory data analysis (e.g., dealt with outliers, null values, and multicollinearity) and conducted feature engineering to transform some of our datetime and categorical variables into meaningful predictors through date extraction and one-hot encoding, respectively. Additionally, to get a preliminary look at feature importance, we have split our data into training and test sets and employed a logistic regression classifier to learn the key predictors of Kickstarter campaign failures.

---

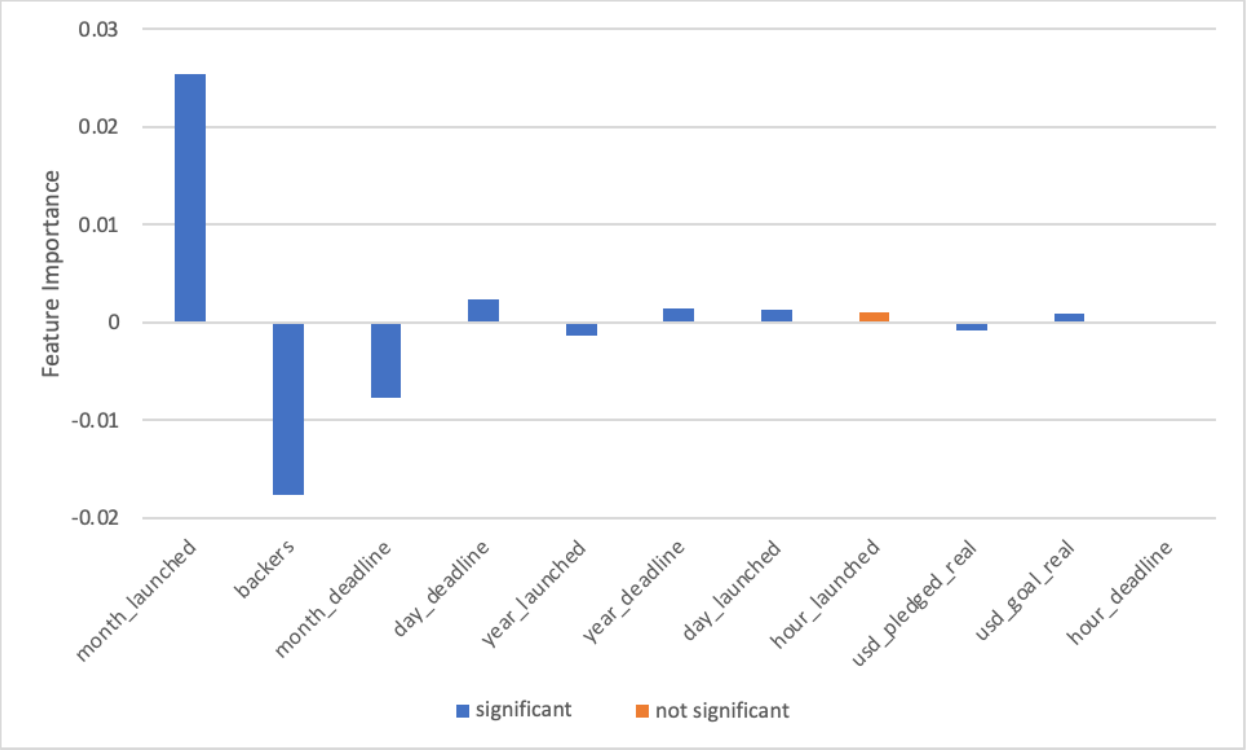
[1] Number of backers, amount pledged by crowd, fundraising goal, category of campaign, location, etc.

[2] Kickstarter campaign success [0] or Kickstarter campaign failure [1]

Our model was surprisingly performant, and here is a high-level summary of key metrics we looked at to evaluate it:

Key Metrics	Results
Precision	0.9949
Recall	0.9847
F1-Score	0.9898
AUC	0.9952
Accuracy on test set	0.9869

From a feature importance standpoint, we analyzed the coefficients of our logistic regression in order to get a sense of the direction and strength of each of our predictors' influence on the predicted probabilities of our binary target variable (which is again a feature that reflects Kickstarter campaign success [0] or Kickstarter campaign failure [1]).



As seen in the visual above, there is a significant relationship between all the metrics included in the model and Kickstarter campaign failure at a 99% confidence level, excluding hour_launched.

Month_launched is the strongest predictor of failure, which implies there is seasonality involved with pledging. Similarly, month_deadline is a solid predictor of failure, which reinforces the hypothesis of an existing seasonality and its impact on a project. Year_launched/year_deadline and day_launched/day_deadline do not have as great of importance in predicting failure.

The backers variable has a large negative relationship with Kickstarter campaign failure; the more backers a campaign has, the less likely they are to fail (more likely to succeed).

Model Assumptions and Requirements

Assumptions for Logistic Regression

1. **Linearity**: The relationship between the independent variables and the log odds of the dependent variable is assumed to be linear.
2. **Independence of Errors**: The observations are assumed to be independent of each other.
3. **No Multicollinearity**: The independent variables should not be highly correlated with each other.
4. **Large Sample Size**: A sufficiently large sample size is needed to ensure reliable parameter estimates.

Requirements for Logistic Regression:

1. **Binary Dependent Variable**: Logistic regression is suitable for predicting binary outcomes (e.g. yes/no or success/failure).
2. **Independent Variables**: Need one or more independent variables to predict the probability of the binary outcome.
3. **Data Preprocessing**: The data should be cleaned, missing values handled, and categorical variables encoded appropriately.
4. **Model Estimation**: Logistic regression estimates model parameters using optimization techniques like Maximum Likelihood Estimation.
5. **Model Evaluation**: The performance of the logistic regression model is assessed using metrics such as accuracy, precision, recall, and F1-score.
6. **Interpretation**: The coefficients of the logistic regression model can be interpreted as the impact of independent variables on the log odds of the outcome.

Note: Log odds (logit) is a transformation of probabilities into a linear scale, used in logistic regression to model the relationship between independent variables and binary outcomes. It converts probabilities (0 to 1) to a range from negative to positive infinity, facilitating linear modeling and prediction.

Balancing Classifiers

Balancing a classifier is important when dealing with imbalanced datasets, where one class significantly outnumbers the other class(es). In such cases, the classifier tends to become

biased towards the majority class and may have poor predictive performance for the minority class. This imbalance can render preliminary results unreliable and misleading. Here are the key reasons why we need to balance a classifier:

1. **Biased Learning**: Imbalanced data can lead the classifier to be biased towards the majority class, resulting in high accuracy for the majority class but poor performance on the minority class. The model may simply predict the majority class most of the time, making it ineffective in capturing the patterns and nuances of the minority class.

2. **Misleading Evaluation Metrics**: Accuracy, the commonly used metric for classifiers, can be misleading in imbalanced datasets. A classifier that always predicts the majority class will have a high accuracy, but it does not reflect the model's ability to correctly classify the minority class instances.

3. **Reduced Sensitivity**: Sensitivity (recall) is an essential metric for imbalanced datasets. It measures the ability of the classifier to correctly identify positive instances (minority class). If the model is not balanced, sensitivity for the minority class might be very low, indicating that it fails to capture important positive instances.

Next Steps

We plan to continue improving our model through hyperparameter tuning, feature engineering, and ultimately choose the model that achieves the strongest predictive accuracy. Given that hyperparameter tuning is an iterative process, we will split the data into training, validation, and test sets. We will train the model using the training set, which will build the model’s predictive capabilities. The validation set will be used to compare the various hyperparameter configurations, and we will choose the strongest performer. We will then use the test set as the final evaluation of the selected model’s performance, providing an unbiased assessment of its generalizability.