Kickstarter Project Analysis Pipeline Report

Executive Summary

This report provides a detailed overview of the Kickstarter project analysis pipeline, which consists of two primary components:

1. Data cleaning and preprocessing
2. Feature selection and optimization

The pipeline was designed to transform raw Kickstarter project data into a format suitable for machine learning models that can predict project success. The preprocessing phase focuses on handling missing values, creating informative features, preventing data leakage, and encoding categorical variables. The feature selection phase identifies the most relevant predictors through multiple complementary techniques, ultimately delivering an optimized feature set that balances model performance with simplicity.

1. Data Preprocessing Pipeline

1.1 Overview and Goals

The preprocessing pipeline transforms raw Kickstarter project data into a clean dataset ready for modeling. Key objectives include:

- Handling missing values appropriately
- Engineering informative features
- Preventing data leakage
- Encoding categorical variables effectively
- Producing a consistent numeric dataset for modeling

1.2 Preprocessing Steps

Step 1: Missing Values Analysis and Imputation

The pipeline begins by identifying and addressing missing values:

- Detailed analysis of missing value patterns and percentages
- Separation of columns by data type (numeric, categorical, datetime)
- Implementation of multiple imputation methods:
  - Simple imputation (median for numeric, mode for categorical)

  - K-Nearest Neighbors (KNN) imputation for numeric features
  - Multiple Imputation by Chained Equations (MICE)
- Comparison of imputation methods to select the most appropriate approach (KNN selected as final method)

Step 2: Feature Engineering (Streamlined)

The pipeline carefully creates informative features while avoiding redundancy and multicollinearity:

Temporal Features:

- Campaign duration (days between launch and deadline)
- Weekend flags for important dates (e.g., deadline_is_weekend)

Goal-related Features:

- Log transformation of goal amount (goal_log)

Category-based Features:

- Success rates by main category (category_success_rate)

Geographic Features:

- Country-level success rates (country_success_rate)

Text Features:

- Name to blurb length ratio (name_blurb_ratio)

Additional Features:

- Same-day launch indicator (projects created and launched on the same day)

Step 3: Data Leakage Prevention

The pipeline implements rigorous safeguards against data leakage:

- Systematic identification and removal of post-campaign features:
  - Direct leakage features (pledged, backers_count, usd_pledged)
  - State change features (state_changed_at and related columns)
  - Post-success features (spotlight, staff_pick)
- Detection of indirect leakage through correlation analysis with post-campaign metrics
- Documentation of all leakage prevention measures

Step 4: Categorical Encoding

The pipeline employs a selective encoding approach based on feature characteristics:

- Cardinality-based encoding strategy:
  - One-hot encoding for low-cardinality features (<10 unique values)
  - Label encoding for high-cardinality features (≥10 unique values)
- Boolean features converted to integers (0/1)

Step 5: Final Processing

The pipeline concludes with several finalization steps:

- Creation of a binary target variable (1 for successful, 0 for failed projects)
- Removal of original categorical columns that have been encoded
- Conversion of datetime columns to numeric (days since reference date)
- Final missing value check and remediation
- Calculation of feature correlations with the target
- Export of the cleaned, processed dataset

1.3 Output

The preprocessing pipeline produces a clean, numeric dataset with:

- No missing values
- Informative, non-redundant features
- No data leakage
- Properly encoded categorical variables
- A binary target variable for classification

2. Feature Selection Pipeline

2.1 Overview and Goals

The feature selection pipeline identifies the most predictive features using multiple complementary techniques. Key objectives include:

- Reducing dimensionality to improve model performance
- Identifying the most important predictors of project success

- Eliminating redundant or irrelevant features
- Creating an optimal feature subset for final modeling

2.2 Feature Selection Methods

Method 1: Correlation-based Selection

This approach identifies and removes multicollinearity among features:

- Construction of a correlation matrix among all numeric features
- Identification of feature pairs with high correlation (threshold = 0.7)
- For each correlated pair, retention of the feature with stronger correlation to the target
- Visualization of the correlation structure via heatmap

Method 2: Statistical Feature Selection

This method applies statistical tests to measure feature relevance:

- ANOVA F-value selection:
  - Measures the relationship between each feature and the target
  - Ranks features by F-score and p-value
- Mutual Information selection:
  - Captures non-linear relationships between features and target
  - Ranks features by mutual information score
- Visualization of top features from both methods
- Identification of features consistently selected by multiple statistical approaches

Method 3: Model-based Feature Selection

This technique leverages the Random Forest algorithm's built-in feature importance:

- Training of a Random Forest classifier on the full feature set
- Extraction of feature importance scores
- Selection of top features that collectively account for >80% of cumulative importance
- Evaluation of model performance using the selected feature subset
- Comparison against the full-feature model to verify minimal performance impact

Method 4: Recursive Feature Elimination

This systematic approach progressively removes the least important features:

- Utilization of Recursive Feature Elimination with Cross-Validation (RFECV)
- Logistic Regression with L1 penalty as the base estimator
- ROC-AUC as the optimization metric
- Cross-validation to determine the optimal number of features
- Visualization of performance across different feature counts

2.3 Ensemble Feature Selection

The pipeline combines insights from all selection methods:

- Construction of a voting mechanism across methods
- Counting of "votes" for each feature (appearances across methods)
- Selection of features appearing in at least two methods
- Visualization of top features by vote count

2.4 Performance Evaluation

The pipeline rigorously evaluates different feature subsets:

- Consistent train/test split for fair comparison
- Random Forest classifier as the evaluation model
- ROC-AUC as the performance metric
- Comparison of all feature subsets (correlation-based, statistical, model-based, RFE, ensemble, all features)
- Identification of the best-performing feature subset

2.5 Output

The feature selection pipeline produces:

- A ranked list of features by importance
- An optimal feature subset balancing performance and simplicity
- Performance metrics for different feature selection strategies
- Visualizations of feature importance across methods

3. Key Findings and Recommendations

3.1 Most Important Features

Based on the ensemble selection approach, the following feature categories emerged as most predictive:

1. **Project Goal Characteristics:**
   - Log-transformed goal amount
   - Goal USD adjusted
2. **Category and Geographic Patterns:**
   - Category success rates
   - Country success rates
3. **Campaign Structure:**
   - Campaign duration
   - Launch timing features
4. **Project Presentation:**
   - Text length and efficiency metrics

3.2 Recommendations for Kickstarter Project Creators

The analysis suggests several strategies for improving project success:

1. **Set Realistic Goals:** The goal amount is consistently one of the strongest predictors. Lower, more achievable goals correlate with higher success rates.
2. **Choose Categories Strategically:** Some categories have significantly higher success rates than others. Research category performance before launching.
3. **Optimize Campaign Duration:** The data suggests an ideal campaign duration exists. Excessively short or long campaigns may reduce success probability.
4. **Craft Effective Presentations:** Text metrics emerged as important predictors. Clear, concise project descriptions appear to influence success.
5. **Consider Geographic Factors:** Success rates vary by country. Understanding regional patterns can inform launch strategies.

3.3 Technical Recommendations

For future improvements to the pipeline:

1. **Feature Engineering Refinement:**
   - Consider deeper text analysis of project descriptions
   - Explore network effects between creator history and project success

2. **Model Ensemble:**
    - Use the identified optimal feature subset with multiple model types
    - Create an ensemble classifier for improved prediction accuracy
3. **Temporal Validation:**
    - Implement time-based validation to better simulate real-world prediction scenarios
    - Test for concept drift in feature importance over time
4. **Interpretable Models:**
    - Develop interpretable models to provide actionable insights to creators
    - Create visualization tools for project creators to assess their project's success probability

4. Conclusion

The Kickstarter project analysis pipeline successfully transforms raw project data into an optimized dataset for predictive modeling. The preprocessing phase handles missing values, engineers informative features, prevents data leakage, and properly encodes categorical variables. The feature selection phase identifies the most predictive features through multiple complementary techniques, ultimately delivering an optimal feature subset that balances model performance with simplicity.

The identified predictors offer valuable insights for both project creators and the platform itself. By understanding the factors that drive project success, creators can optimize their campaigns, and the platform can better support projects with the potential for success.