Individual Project

Summary Report

Éloi Dallaire (260794674)

Desautels Faculty of Management, McGill University

INSY 446: Data Mining for Business Analytics

Professor Elizabeth Han

December 10, 2023

1. Understanding the Data

After gaining an early understanding of the different types of variables in the dataset, we process to a rapid pre-cleaning before diving into our exploratory data analysis (EDA). This step included removing all the cancelled or suspended projects and handling any duplicates. Moreover, the entirety of the missing values was coming from the 'category' variable, nearly 10% of the total observations. By fear of losing any predictive power and because of that high proportion, we rather reassigned them to a proper 'missing' category. This could potentially lead to the identification of insightful and relevant trends behind those missing values. The EDA of both categorical and numerical variables revealed trends, distributions, and potential outliers. Proportional tables exposed intriguing relationships, such as 'staff_pick' significantly influencing success.

2. Pre-processing

We streamlined the dataset by removing identifiers, variables highly correlated to the target variable, unary predictors and those realized after campaign launch. We also dropped variables related to the name length, as they ended up being very uninformative since never matching the actual project's name. A total of 22 predictors were dropped. We slightly reformatted variables in a 'datetime' format and 'goal', to reflect them on a consistent base (respectively delta time from the oldest project & USD currency). To help with data manipulation and clarity, all binary variables were transformed to Boolean and categorical variables were one-hot encoded. We kept the data unstandardized, as the algorithms used for classification can support such predictors. Lastly, we split into training and testing sets, preserving the stratification of project states, avoiding disproportionate distribution of succeed and failed projects.

3. Classification Model

Base Model, Validation Method & Performance Metrics

To establish a baseline, a Logistic Regression model was built, providing a benchmark for subsequent models. Cross-validation with 5 folds was used throughout the whole process, to ensure consistent comparison of the model's performance. Evaluation metrics included were accuracy, precision, recall, and F1 score. On one hand, accuracy was chosen as the main metric, as it aligned with our project's objective of providing a measure of overall correctness in predicting Kickstarter project outcomes. Yet, it’s noted that in imbalanced datasets, additional metrics provide a more nuanced evaluation.

Building Models, Hyperparameter Tuning & Outliers Removal

In order to aim for the best accuracy attainable while still keeping a good level of interpretability and cost-efficiency, we decided to built both a Random Forest and a Gradient Boosting model. After that, we used the GridSearchCV algorithm to help find the optimal values of the following hyperparameters: ‘n_estimators’, ‘max_features’, ‘min_samples_split’ and ‘min_samples_leaf’. We followed a binary search type of approach, starting with a wide range and narrowing down towards the optimal values. To prevent risk of overfitting, we limited the ‘max_features’ to 9. Concerning the balance between model complexity and generalization, an evident sign of overfitting that we looked for was potential performance discrepancies between training and testing data. An efficient model must be robust enough to learn from trends in the training stage while not generalizing too much to it. Ultimately, the tuning processes helped improve most of the measures by around 1-2% for both models (see appendix for comprehensive results). The IsolationForest algorithm with a 2% contamination rate was applied to identify and exclude outliers from both training and test sets. In total, 237 observations were removed. Their removal played a pivotal role in enhancing both model’s resilience, robustness and reliability. Overall, the performance metrics from both models were improved by around 0.10%.

Feature Selection

A critical phase involved feature selection, where variables were scrutinized for relevance. For this step, the objective was to reduce the complexity of a model by selecting fewer features. This will highly reduce the risk of overfitting and help gain a better accuracy on unseen data. The feature importance attribute of both models was employed to rank all the available predictors. Naturally, feature selection decreased the accuracy of our models with the trained data. The key here was to find the best balance between getting rid of a maximum number of predictors while limiting the impact on the model’s performance. In the end, by keeping a feature importance threshold of 4.5%, we selected the following

6 predictors: ‘staff_pick’, ‘category_Web’, ‘category_Software’, ‘goal_usd’, ‘create_to_launch_days’ and ‘launched_at’. The decrease in accuracy on both training and test dataset was limited to 1.7% and 3% respectively for the Gradient Boosting model.

Final Model Selection

At each stage, cross-validation and thorough evaluation were integral. Comparative analysis of Random Forest and Gradient Boosting performance allowed us to discern the most effective model, informing our final selection. We also visualized both model’s predictive power through ROC curves and AUC analysis, demonstrating its ability to discriminate between successful and failed campaigns. In conclusion, the Gradient Boosting model was selected as our final model with a final accuracy of 75% on the test data.

4. Clustering Model

To complement the earlier classification model, we explored clustering using PCA (reduced to 2 dimensions) and tested various algorithms, such as K-Means, K-Prototypes, and DBSCAN, with different features and data composition. Ultimately, K-Means proved the most suitable method due to uniform and equally dense cluster shapes. For determining the optimal number of clusters, Elbow and Silhouette methods converged on 4 or 6 clusters as optimal. Despite poor separation and low cohesion in the obtained clusters, valuable insights were derived. Cluster 1 & 3 projects showed a higher success rate when launched in earlier months and days of the week. Cluster 2 exhibited older projects with a shorter creation-to-launch period, indicating a lower success rate, often paired with a missing or very short blurb. Notably, clusters revealed a trend linking higher USD static rates (e.g., EUR or GBP) with a better project success rate.

5. Conclusion

This comprehensive analysis of Kickstarter campaigns provides valuable insights for both project creators and backers. Predictive models offer a proactive approach to project success, while clustering opens avenues for targeted marketing and audience understanding. Future work involves refining clustering and deploying predictive models for real-time campaign monitoring.

6. Appendix

Classification – Performance Evaluation and Model Comparison

Models	Accuracy	Precision	Recall	F1 Score	Avg CV Accuracy
1. Logistic Regression - Base model	66.01%	66.67%	39.74%	69.00%	65.57%
2.1 Random Forest - Initial	77.02%	72.42%	52.52%	60.88%	76.52%
2.2 Random Forest - After Tuning Hyperparameters	77.42%	71.73%	55.63%	62.66%	77.20%
2.3 Random Forest - Without Outliers	77.50%	72.34%	54.95%	62.46%	77.25%
2.4 Random Forest - After Feature Selection	71.90%	60.53%	50.26%	54.92%	72.15%
3.1 Gradient Boosting - Initial	77.45%	71.46%	56.23%	62.94%	77.48%
3.2 Gradient Boosting - After tuning hyperparameters	77.88%	70.94%	59.34%	64.62%	77.65%
3.3 Gradient Boosting - Without Outliers	78.00%	71.31%	59.26%	64.73%	77.87%
3.4 Gradient Boosting - After Feature Selection	**75.01%**	**65.20%**	**57.09%**	**60.88%**	**76.14%**