Emily Wu 261148026

INSY 662 Data Mining Individual Project

Classification Model

The data preprocessing included removing rows with missing values (1,409 in total), discarding post-launch features like spotlight and staff pick, encoding categorical variables into dummies, and standardizing the features for the K-Nearest Neighbors (KNN) model.

For feature selection, I utilized lasso regression, random forest importance scores, and principal component analysis. The impact of the selected features was evaluated using both KNN and Random Forest models. In tuning the KNN model, I tested a neighbor range of 15-25 and found that 18 neighbors provided the highest accuracy. Utilizing lasso for feature selection marginally increased the KNN model's accuracy to 0.696, while PCA with nine components achieved an accuracy of 0.685. The optimal number of components in PCA was identified via the elbow method, where further increase in components did not significantly contribute to variance explanation. Lasso regression, implemented through LassoCV, determined an optimal alpha of 0.001, and highlighted less important factors including certain weekdays, the 'wearables' category, and specific countries (CH, FR, DE, etc.).

In the Random Forest model, accuracy improved from 0.740 to 0.746 after applying lasso-based feature selection, with the best results obtained using 200 trees and 35 features. Additional hyperparameter adjustments, such as maximum tree depth and minimum samples per split or leaf, did not enhance the model accuracy.

Among various models experimented, including KNN, Random Forest, Gradient Boosting, and Artificial Neural Networks (ANN), the Random Forest with lasso-selected features outperformed others. Gradient Boosting reached an accuracy of 0.712 with the best setup of one feature per

split and 100 estimators. The ANN, configured with a single 36-node layer, a maximum of 1,000 epochs, and a logistic activation function, yielded an accuracy of 0.677.

Ultimately, the Random Forest model with features selected by lasso regression proved to be the most accurate, achieving an accuracy of 0.746.

Clustering Model

To implement an unsupervised clustering model, both K-Means and DBSCAN were evaluated since both algorithms can identify structures or patterns in data without needing any labels and are scaled to work with large datasets. However, DBSCAN proved unsuitable for the dataset. Initially, I attempted to incorporate all numerical variables along with "category" and "state." However, this approach limited the ability to effectively evaluate the k-prototype model performance, as metrics like silhouette score and inertia are not applicable in this context. Consequently, in the final model, I opted for an exclusively numerical variable approach using K-Means. Following the rule of thumb, cluster numbers ranging from 2 to sqrt(n/2) were tested. This revealed that a four-cluster model (n_cluster=4) achieved a commendable inertia of 624,340,673,496,471 and a silhouette score of 0.970452349. Notably, in one cluster, row 12197 was an isolated data point. Removing this row and varying the number of clusters led to a silhouette score of 0.968 and a pseudo F-score of 45,578.6052156904. Moreover, the inertia markedly reduced to 454,285,153,014,871 from the earlier 625,707,333,169,191.8 noted with three clusters (n_cluster=3).

The key insights from this unsupervised clustering model are intriguing. Projects with higher funding goals (goal_usd) tend to attract fewer backers. Additionally, higher project goals

correlate with lower actual pledged amounts. It was also observed that projects with shorter intervals from creation to launch generally have higher funding goals.

In summary, these findings suggest that for enhancing project success rates on platforms like Kickstarter, the project initiators should consider setting more realistic funding goals. Excessively high goals may appear unattainable, potentially diminishing the confidence and willingness of potential backers to support the project. This perception of unfeasibility could adversely impact the backing behavior, as indicated by the model.

Fig1. Inertia and Silhouette Scores of clustering model

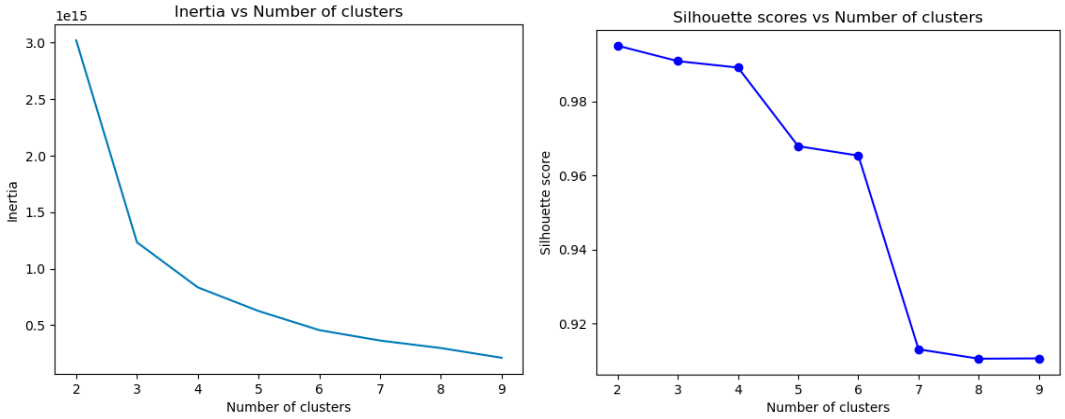

Fig 2. Key findings of clustering model