Kickstarter is a funding platform where their main goal is to transform ideas into projects. Each project is created and crafted by the person behind it where once they're ready, they can launch their project to the community. Every project creator sets their funding goal and deadline, and if the project gains enough interest, people can pledge money to make it happen. This project is divided into 3 main parts; regression, classification and clustering which will be covered in detail.

Data Preprocessing

As a starting point, I imported the clean dataset of 18568 observations and 45 columns and kept only the observations where the project state is either successful or failed, reducing the amount of observations to 15685. The following tables shows the predictors that were dropped:

Predictor	Reason Behind Removal
Project_ID, name	Used for identification, have no effect on the model
Goal, static_usd_rate	A new variable was created that multiplies both for a more scalable comparison
Pledged	After the project was submitted
State	After the project was submitted (For regression)
Disable_communication	After the project was submitted
Currency	Highly correlated to country
Deadline, State_changed_at, created_at, launched_at	Their respective dummies were used
Backers_count	After the project was submitted
Spotlight	After the project was submitted
Staff_pick	After the project was submitted
Name_len, blurb_len, Name_len_clean, blurb_len_clean	Due to feature selection they were dropped as they decreased model's performance (they were kept for classification)
State_changed_at_weekday, month, day, year, hour	After the project was submitted
Launch_to_state_change_days	After the project was submitted
Usd_pledged	After the project was submitted (for classification task)

After dropping the previous predictors, dummies were created for the following categorical variables: country, category, deadline_weekday, created_at_weekday, created_at_yr,deadline_yr, deadline_month, created_at_month. The next part of the process involved the removal of anomalies using Isolation Forest. In the regression part, it was realized that when removing outliers for the whole dataset, the target variable once plotted its boxplot still had extreme outliers. Hence, two different Isolation Forest models were built for the target variable and for the predictors respectively, reducing the total amount of observations to 13793. As for the clustering part, anomaly detection was performed on my 3 predictors. To further reduce the number of input variables, feature selection was performed for both regression and classification tasks, using Lasso and Logistic feature selection techniques.

Model Selection

For both regression and classification tasks, the optimal model was selected based on the least MSE and highest accuracy using cross-validation. Different models were tested such as Lasso, Ridge, Multiple Linear Regression, KNN, ANN, Random Forest and Gradient Boosting. However, in both models Gradient Boosting was selected. The optimal hyperparameters were selected by looping through each hyper parameter. The tables below represent the results:

Regression:

Technique	**Ridge**	**Lasso**	**Multiple Linear Regression**	**KNN**	**ANN**	**Random Forest**	**Gradient Boosting**
MSE (in billion)	1.322	1.324	1.321	1.354	1.31	1.217	1.206

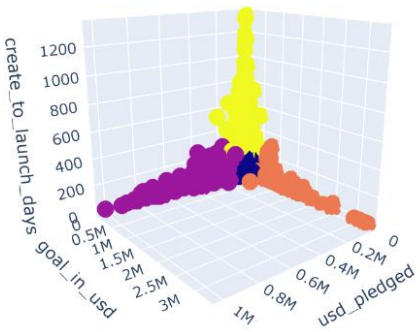
Classification:

Technique	**KNN**	**ANN**	**Random Forest**	**Gradient Boosting**
CV Score	0.6994	0.6719	0.7473	0.758

	Regression	Classification
Max features	Auto	Auto
Max depth	2	6
Minimum samples split	42	9
Min Samples leaf	13	3
N_estimators	90	250

From a business perspective, the prediction model requires more variables to provide a higher and more reliable model through for example an industry analysis of each category. As for the classification model it can be recommended due to its ability to classify the state of a project at a 75.6% accuracy.

For the clustering model, it was performed on USD_pledged, Goal in USD and Create to lunch days. After dropping NAs and removing anomalies using Isolation Forest, K-Means Clustering was selected, where using elbow method the optimal number of clusters was 4. As we can see from the 3D scatter using Plotly, it is easy to visualize our 4 clusters. Yellow Cluster represents creators who took too long to launch with low goals and ended with low USD_pledged. Purple cluster represents creators who had low to intermediate goals and time to launch but exceeded their own expectations and ended up with a high USD_pledged. Orange Cluster represents creators with intermediate-high goals, low-intermediate time to launch and ended with low USD_pledged. Finally, the blue cluster represents creators with low goals, launched quickly and ended with very low USD_pledged.



The model was then tested using silhouette score (0.775), Calinski-Harabasz score (8551.06) and p-value (1.11e-16), meaning that our data was well clustered.