Stats 504 Assignment 3: Kickstarter Success

Introduction

Crowdfunding is the practice of funding a project or venture by raising money from a large number of people, typically via the internet. This report aims to serve as an informational guide for start-ups attempting to raise capital on the Kickstarter crowdfunding platform, detailing what constitutes a successful project. The report addresses specific questions of interest such as the influence of the goal amount in the Kickstarter project, number of unique backers, and geographic location of the project. Analysis reveals that the goal amount and number of backers are extremely powerful in predicting whether or not a project is successful, whereas the country the project is hosted in plays little to no effect (especially with the USA) . Following the analysis, the report also provides information regarding expected probability of success given varying scenarios for the project.

Methods

The goal of the analysis is to help depict and understand the features of a Kickstarter project that influence whether or not it will be successful (i.e raise the amount of money pre-determined as the goal), and provide concrete evidence in identifying other useful information for the client. Since this boils down to associating each covariate with an idea of how it sways the probability of success, logistic regression was chosen as the model of choice. Logistic Regression provides, for each variable included in the model, a coefficient which affects the probability of the response variable (i.e Success or Failure of project), which is precisely what is desired from this analysis. As with any useful model, logistic regression also comes with its share of complexities and limitations. For this dataset, in the presence of "downstream" data variables, such as the amount of money pledged, logistic regression will fail to work as the dataset suggests an overly trivial rule of "if the pledged amount is greater than the predefined goal amount, then the project will succeed", which cannot be applied in a meaningful real world use-case. This is known as perfect separation in the statistical community. After careful consideration, data was selected so as to avoid the perfect separation problem, and build a usable model.

As a secondary analysis, which could aid in answering the client's questions, a decision tree was also fit to the data. A decision tree is quite self-explanatory, and represents a sequence of questions about the data values of interest in order to classify the project as a success or failure. The downside of using a decision tree is that it does not provide any information about the individual influence of each variable, but that is covered by the Logistic Regression. The purpose of the decision tree is purely for quick high level interpretation by the client. The results are presented in the following section.

Results

The client provided a URL to the data which was hosted on the internet at the data.world website. This data represented past projects on Kickstarter along with information about the project, as well as whether or not it was successful in raising enough money. Each row of the data represents a single Kickstarter project and there are 20632 such records. For each project, there was a multitude of information most of which proved fairly unusable. If the data was text based (eg: blurb, project description, etc) or date based (launch, deadline, etc) it was discarded from the analysis. This was primarily because there was no need for any sort of text or time series analysis. There were a few other columns in the data that were mostly missing, so those were dropped as well. Finally, if any data had the potential to be a downstream variable, it was excluded from the analysis. There were a few rows which represented "live" campaigns, i.e campaigns that were ongoing and whose resolution on success/failure was unknown. Since this was a small proportion, all such rows were dropped from the analysis. The regression analysis was finally performed on 20124 different projects with each project having 4 different features which are further described in Table 2.

Feature	Median (IQR) / Percentage
Goal (USD)	13,488 (4,000 - 45,000)
Project Status	
Success	6018 (29.9%)
Failed	11416 (55.3%)
Suspended	230 (1.1%)
Cancelled	2460 (11.9%)
Live	508 (2.5%)
Number of Backers (successful projects)	105 (39 - 380)
Duration from Launch to Deadline (Days)	30 (30 - 40)
Based in USA	13835 (68.7%)
Category	
Web	3267 (17.8%)
Hardware	3202 (17.5%)
Software	2593 (14.2%)
Gadgets	2275 (12.4%)
Plays	1161 (6.3%)
Apps	1044 (5.7%)
Wearables	938 (5.1%)
Musical	775 (4.2%)
Misc	3023 (16.5%)

Table 1: Baseline table indicating summary statistics of data relevant to the analysis

Multiple variations of regression models were fitted, but only the (subjective) best one is discussed here. The table below describes the effects of various features of the project on its rates of success.

Covariate	**Change in % for Odds of Success**	**p-value**
Goal	- 0.01	<0.0001
Backers	+ 1.6	<0.0001
Based in USA	+ 5.0	0.25
Launch to Deadline (days)	- 1.8	<0.0001

Table 2: Percentage change expected with unit increase of covariates

As the table suggests, goal amount and the number of backers, were the features that most influenced the success of a project, while being based out of the USA was deemed as mostly irrelevant by the model. The duration of the project, in terms of days between launch and deadline was another statistically significant term in the model. To interpret the coefficients from the logistic regression, they need to be exponentiated in order to obtain the odds ratio. The odds ratio can be interpreted as follows. For example, with all other features held constant, having one more backer increases the odds of success by 1.6%. Other features can be similarly interpreted from the table.

It can be argued that the number of backers is not something the client has control of, as it is just the number of unique people on the platform who decided to pledge money to the project. Therefore, using this data point in order to provide information about the probability of success might be questionable. However, it is still fitted in the model and it proves to be significant. Another reason for not excluding the number of backers is that the average pledge amount per backer was analyzed. The median amount a backer pledges is around $57. Another interpretation of the backers count now emerges, in terms of its relationship with the median pledge amount. With the reasonable assumption that each backer pledges the median amount, the count of backers now can also indicate whether or not the goal amount will be reached. Due to these reasons, the model was built with the inclusion of the number of backers as a variable.

In order to shed additional light for the client, the model can be used to compute the expected probability of success given a set of scenarios under which the project is launched. The following table presents some realistic scenarios for the client to consider. With respect to the number of backers, while the client specifically enquires about having at least 1000 backers, it is insightful to look at lower numbers as it seems the client underestimates the amount each backer tends to pledge. The client is likely to achieve the goal amount with a lower number of backers.

Numbers of Backers	Launch to Deadline Duration (Days)	% of Success
50	7	24.09%
100	14	38.43%
200	30	69.96%
300	60	87.08%
1000	90	99.99%

Table 3: Chance of success for a project based in USA, for a goal amount of $25,000

Finally, as a secondary analysis, a decision tree was fitted to the data in order to aid the client with more interpretability, as that is one of the main advantages of a decision tree. The results are depicted in the figure below. For each stage of the decision tree, following the left arrow represents having a value lesser than or equal to the value depicted by the arrow head on the bar graph. Similarly, the right arrow takes the decision path of having a value greater than the one marked with the arrow head.

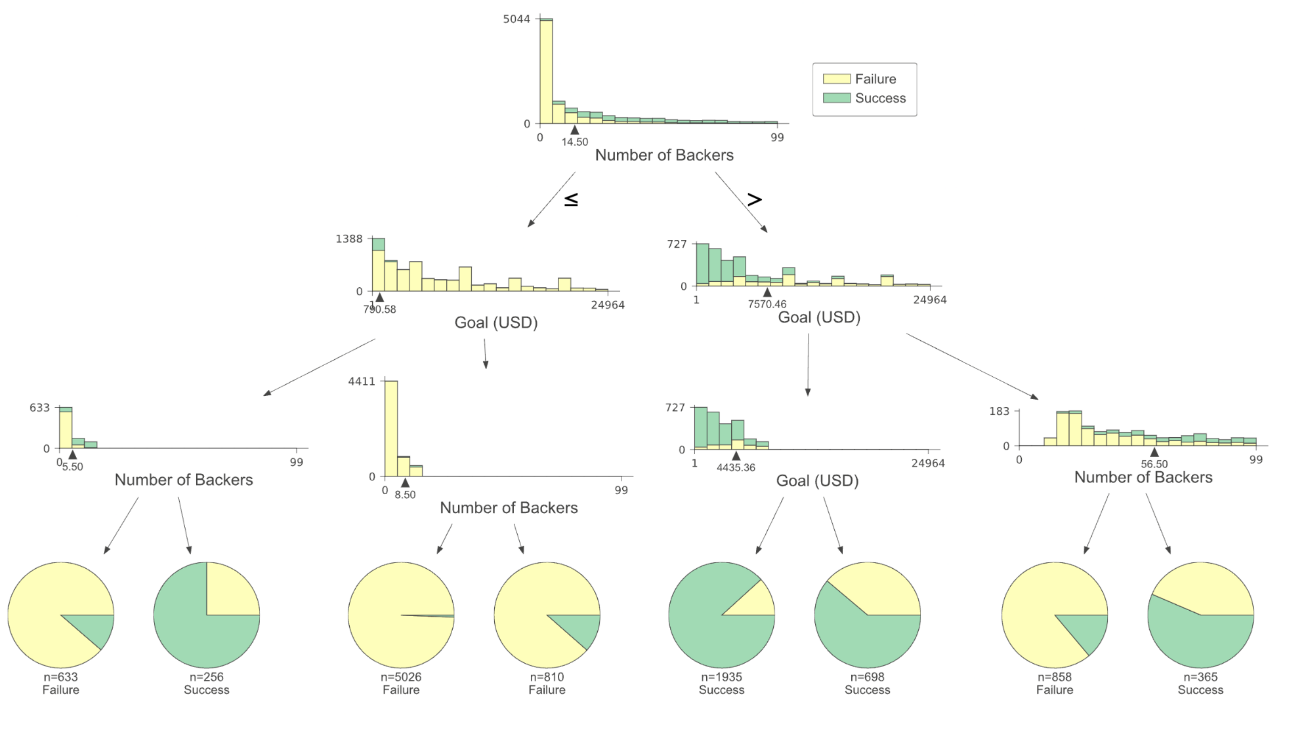


Figure 1: Decision tree depicting the influence of the number of backers and goal amount. A higher goal amount generally leads to a higher number of failures, while having more backers is a common trend in all projects that were successful.

Conclusion

This report presents the results of an analysis performed on Kickstarter project information, with the objective of trying to understand what features influence success on the platform. It also addresses the key questions posed by the client regarding specific features like the goal amount they are looking to raise ($25,000), and minimum amount of backers they are expecting (1000). Both these variables are deemed significant by the model, and they influence success probabilities by -0.01% and 1.6% respectively. While these results are expected to be useful, it must be noted that the model comes with a fair share of limitations. As with any regression modeling, the analysis can only yield information about associations between data, and it is unwarranted to make assumptions about causal relationships. Since a Kickstarter project is a very broad and abstract subject, the data points being used to describe it may display associations owing to other causal relationships not represented in the data. However, it is expected that these results will still guide the client in launching their Kickstarter campaign with the highest possible chance of success.

Appendix

In [ ]:

```

```

In [312…

```
from matplotlib import pyplot as plt
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
```

In [313…

```
import statsmodels.api as sm
import patsy
```

Read in the data

In [349…

```
import pandas as pd
df = pd.read_csv('https://query.data.world/s/lxnrwj5w73bsigranne42td54f54sm', in
```

```
/opt/anaconda3/lib/python3.8/site-packages/IPython/core/interactiveshell.py:316
5: DtypeWarning: Columns (29,30,31,32) have mixed types.Specify dtype option on
import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [350…

```
df["state"].value_counts(normalize=True)
```

Out[350…

```
failed        0.553315
successful    0.291683
canceled      0.119232
live          0.024622
suspended     0.011148
Name: state, dtype: float64
```

Drop Unnecessary Columns

In [351…

```
# drop columns that we don't think will be useful
df["goal"] = df["goal"] * df["static_usd_rate"]
df["isUS"] = df["country"].apply(lambda c: 1 if c == "US" else 0).astype("catego
df["pledge_per_backer"] = df["usd_pledged"] / df["backers_count"]
df = df[df["state"] != "live"]
df = df.drop(
    [
        "id",
        "photo",
        "name",
        "blurb",
        "pledged",
        "slug",
        "state_changed_at",
        "creator",
        "location",
```

```
        "profile",
        "urls",
        "source_url",
        "friends",
        "is_starred",
        "is_backing",
        "permissions",
        "name_len",
        "name_len_clean",
        "state_changed_at_weekday",
        "created_at_weekday",
        "state_changed_at_day",
        "state_changed_at_yr",
        "state_changed_at_hr",
        "created_at_weekday",
        "created_at_yr",
        "created_at_hr",
        "create_to_launch",
        "created_at_month",
        "blurb_len_clean",
        "blurb_len",
        "currency_symbol",
        "currency_trailing_code",
        "created_at",
        "create_to_launch_days",
        "staff_pick",
        "spotlight",
        "usd_pledged",
        "state_changed_at_month",
        "created_at_day",
        "launch_to_state_change",
        "launch_to_state_change_days",
        "launched_at",
        "launch_to_deadline",
        "static_usd_rate",
        "country",
        "currency",
        "deadline",
        "state",
        "USorGB",
        "TOPCOUNTRY",
    ], axis=1)
```

Investigate average pledge amount per backer

In [352…

```
df["pledge_per_backer"].describe()
```

Out[352…

```
count    17318.000000
mean       108.380333
std        197.749285
min          0.471178
25%         25.135714
50%         57.142857
75%        116.355251
max       5000.500000
Name: pledge_per_backer, dtype: float64
```

In [353…

```
df.columns
```

Out[353…

```
Index(['goal', 'disable_communication', 'backers_count', 'category',
       'deadline_weekday', 'launched_at_weekday', 'deadline_month',
       'deadline_day', 'deadline_yr', 'deadline_hr', 'launched_at_month',
       'launched_at_day', 'launched_at_yr', 'launched_at_hr',
       'launch_to_deadline_days', 'SuccessfulBool', 'LaunchedTuesday',
       'DeadlineWeekend', 'isUS', 'pledge_per_backer'],
      dtype='object')
```

In [354…

```
df["category"] = df["category"].astype("category")
```

EDA

In [355…

```
df.head()
```

Out[355…

	goal	disable_communication	backers_count	category	deadline_weekday	launched_a
0	1500.0000	False	0	Academic	Friday	
1	500.0000	False	0	Academic	Friday	
2	100000.0000	False	5	Academic	Thursday	
3	5000.0000	False	0	Academic	Monday	
4	3591.2846	False	17	Academic	Monday	

In [356…

```
df.corr()
```

Out[356…

	goal	disable_communication	backers_count	deadline_month	dea
goal	1.000000	-0.003383	0.006229	0.000255	
disable_communication	-0.003383	1.000000	0.004403	-0.003880	
backers_count	0.006229	0.004403	1.000000	0.004340	
deadline_month	0.000255	-0.003880	0.004340	1.000000	
deadline_day	-0.013641	0.015332	-0.009020	0.016969	
deadline_yr	0.002396	0.035147	-0.018983	-0.213964	
deadline_hr	0.001606	-0.007863	-0.025546	-0.019458	
launched_at_month	0.001955	-0.006452	0.008554	0.532651	
launched_at_day	0.003042	0.015060	0.007889	0.027054	
launched_at_yr	0.000516	0.035762	-0.020998	-0.105063	
launched_at_hr	0.006108	0.005346	-0.049709	-0.027702	
launch_to_deadline_days	0.045165	0.010702	0.021530	-0.026715	
SuccessfulBool	-0.033666	-0.070231	0.194228	0.006702	
LaunchedTuesday	-0.000616	0.009506	0.028621	0.022543	
DeadlineWeekend	-0.007380	0.003879	-0.006962	-0.020890	

	goal	disable_communication	backers_count	deadline_month	dea
pledge_per_backer	0.013624	-0.003644	0.004482	0.020873	

Logistic Regression Model fitting

In [357…

```
y, X = patsy.dmatrices("SuccessfulBool ~  goal  + backers_count   + \
        launch_to_deadline_days + isUS", df, return_type="dataframe")
model0 = sm.Logit(y, X).fit()
print(model0.summary())
print(model0.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.378532
         Iterations 11
                           Logit Regression Results
==============================================================================
Dep. Variable:         SuccessfulBool   No. Observations:                20124
Model:                          Logit   Df Residuals:                    20119
Method:                           MLE   Df Model:                            4
Date:                Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3795
Time:                        21:11:21   Log-Likelihood:                -7617.6
converged:                       True   LL-Null:                       -12277.
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================
===========
                              coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----------
Intercept                  -0.3142      0.068     -4.610      0.000      -0.448
-0.181
isUS[T.1]                   0.0494      0.043      1.145      0.252      -0.035
0.134
goal                    -6.237e-05   1.64e-06    -38.111      0.000   -6.56e-05
-5.92e-05
backers_count               0.0161      0.000     47.262      0.000       0.015
0.017
launch_to_deadline_days    -0.0181      0.002    -10.000      0.000      -0.022
-0.015
================================================================================
===========

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
15245.14489293655
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [358…

```
y, X = patsy.dmatrices("SuccessfulBool ~  goal  + backers_count   + \
       launch_to_deadline_days + isUS  \
         ", df, return_type="dataframe")
model1 = sm.Logit(y, X).fit()
print(model1.summary())
print(model1.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.378532
         Iterations 11
                           Logit Regression Results
==============================================================================
Dep. Variable:          SuccessfulBool   No. Observations:                20124
Model:                           Logit   Df Residuals:                    20119
Method:                            MLE   Df Model:                            4
Date:                 Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3795
Time:                         21:11:28   Log-Likelihood:                -7617.6
converged:                        True   LL-Null:                       -12277.
Covariance Type:             nonrobust   LLR p-value:                     0.000
================================================================================
===========
                              coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----------
Intercept                  -0.3142      0.068     -4.610      0.000      -0.448
-0.181
isUS[T.1]                   0.0494      0.043      1.145      0.252      -0.035
0.134
goal                    -6.237e-05   1.64e-06    -38.111      0.000   -6.56e-05
-5.92e-05
backers_count               0.0161      0.000     47.262      0.000       0.015
0.017
launch_to_deadline_days    -0.0181      0.002    -10.000      0.000      -0.022
-0.015
================================================================================
===========

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
15245.14489293655
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [ ]:

In [359…

```
y, X = patsy.dmatrices("SuccessfulBool ~  goal + backers_count + isUS", df, retu
model2 = sm.Logit(y, X).fit()
print(model2.summary())
print(model2.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.381138
         Iterations 11
                           Logit Regression Results
==============================================================================
Dep. Variable:          SuccessfulBool   No. Observations:                20124
Model:                           Logit   Df Residuals:                    20120
Method:                            MLE   Df Model:                            3
Date:                 Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3752
Time:                         21:11:31   Log-Likelihood:                -7670.0
converged:                        True   LL-Null:                       -12277.
Covariance Type:             nonrobust   LLR p-value:                     0.000
```

```
===============================================================================
=
                      coef    std err          z      P>|z|      [0.025      0.97
5]
-------------------------------------------------------------------------------
-
Intercept          -0.8921      0.038    -23.760      0.000      -0.966      -0.81
9
isUS[T.1]           0.0467      0.043      1.087      0.277      -0.038       0.13
1
goal            -6.403e-05   1.64e-06    -38.970      0.000   -6.72e-05   -6.08e-0
5
backers_count       0.0162      0.000     47.366      0.000       0.015       0.01
7
===============================================================================
=

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
15348.025740553094
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [360…

```python
y, X = patsy.dmatrices("SuccessfulBool ~ goal  + backers_count", df[df["isUS"] =
model3 = sm.Logit(y, X).fit()
print(model3.summary())
print(model3.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.380060
         Iterations 11
                           Logit Regression Results
==============================================================================
Dep. Variable:         SuccessfulBool   No. Observations:                13835
Model:                          Logit   Df Residuals:                    13832
Method:                           MLE   Df Model:                            2
Date:                Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3902
Time:                        21:11:31   Log-Likelihood:                -5258.1
converged:                       True   LL-Null:                       -8622.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
===============================================================================
=
                      coef    std err          z      P>|z|      [0.025      0.97
5]
-------------------------------------------------------------------------------
-
Intercept          -0.8735      0.029    -29.641      0.000      -0.931      -0.81
6
goal            -6.155e-05   1.88e-06    -32.677      0.000   -6.52e-05   -5.79e-0
5
backers_count       0.0162      0.000     40.475      0.000       0.015       0.01
7
===============================================================================
=

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
```

```
quasi-separation. In this case some parameters will not be identified.
10522.26702526179

/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [361…

```
y, X = patsy.dmatrices("SuccessfulBool ~  goal + backers_count    + \
       launch_to_deadline_days + LaunchedTuesday \
         ", df, return_type="dataframe")
model4 = sm.Logit(y, X).fit(maxiter=1000)
print(model4.summary())
print(model4.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.378557
         Iterations 11
                           Logit Regression Results
==============================================================================
Dep. Variable:         SuccessfulBool   No. Observations:                20124
Model:                          Logit   Df Residuals:                    20119
Method:                           MLE   Df Model:                            4
Date:                Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3795
Time:                        21:11:32   Log-Likelihood:                -7618.1
converged:                       True   LL-Null:                       -12277.
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================
===========
                              coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----------
Intercept                  -0.2871      0.063     -4.573      0.000      -0.410
-0.164
goal                    -6.239e-05   1.64e-06    -38.113      0.000   -6.56e-05
-5.92e-05
backers_count               0.0161      0.000     47.338      0.000       0.015
0.017
launch_to_deadline_days    -0.0181      0.002     -9.990      0.000      -0.022
-0.015
LaunchedTuesday             0.0256      0.048      0.527      0.598      -0.069
0.121
================================================================================
===========

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
15246.181237684226

/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [362…

```
y, X = patsy.dmatrices("SuccessfulBool ~  goal  + backers_count   + \
       launch_to_deadline_days + isUS  \
         ", df, return_type="dataframe")
model = sm.Logit(y, X).fit()
print(model.summary())
print(model.aic)
```

```
Optimization terminated successfully.
         Current function value: 0.378532
         Iterations 11
```

```
                              Logit Regression Results
==============================================================================
Dep. Variable:         SuccessfulBool   No. Observations:                20124
Model:                          Logit   Df Residuals:                    20119
Method:                           MLE   Df Model:                            4
Date:                Fri, 07 Oct 2022   Pseudo R-squ.:                  0.3795
Time:                        21:11:33   Log-Likelihood:                -7617.6
converged:                       True   LL-Null:                       -12277.
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================
===========
                              coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
-----------
Intercept                  -0.3142      0.068     -4.610      0.000      -0.448
-0.181
isUS[T.1]                   0.0494      0.043      1.145      0.252      -0.035
0.134
goal                    -6.237e-05   1.64e-06    -38.111      0.000   -6.56e-05
-5.92e-05
backers_count               0.0161      0.000     47.262      0.000       0.015
0.017
launch_to_deadline_days    -0.0181      0.002    -10.000      0.000      -0.022
-0.015
================================================================================
===========

Possibly complete quasi-separation: A fraction 0.11 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
15245.14489293655
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

In [ ]:

```
```

In [396…

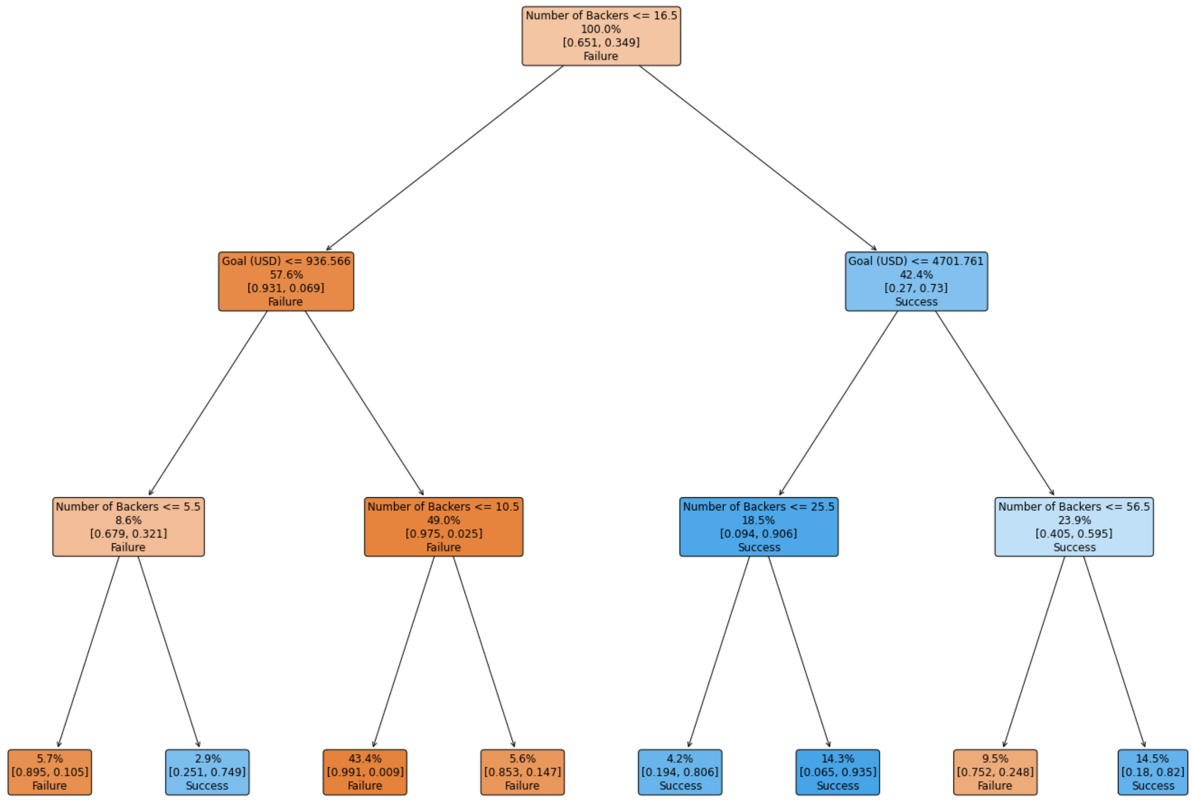
```
y, X = patsy.dmatrices("SuccessfulBool ~  goal    + \
       + backers_count + launch_to_deadline_days + isUS  \
         ", df[(df["backers_count"] < 100) & (df["goal"] < 25000)], return_type=
clf = DecisionTreeClassifier(random_state=1234, max_depth=3)
dt_model = clf.fit(X, y)
```

Decision Tree

In [388…

```
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(clf,
                   feature_names=["", "Is USA?", "Goal (USD)", "Number of Backer
                   class_names=["Failure", "Success"],
                   label="none",
                   impurity=False,
                   proportion=True,
```

```
                     rounded=True,
                     filled=True)
```

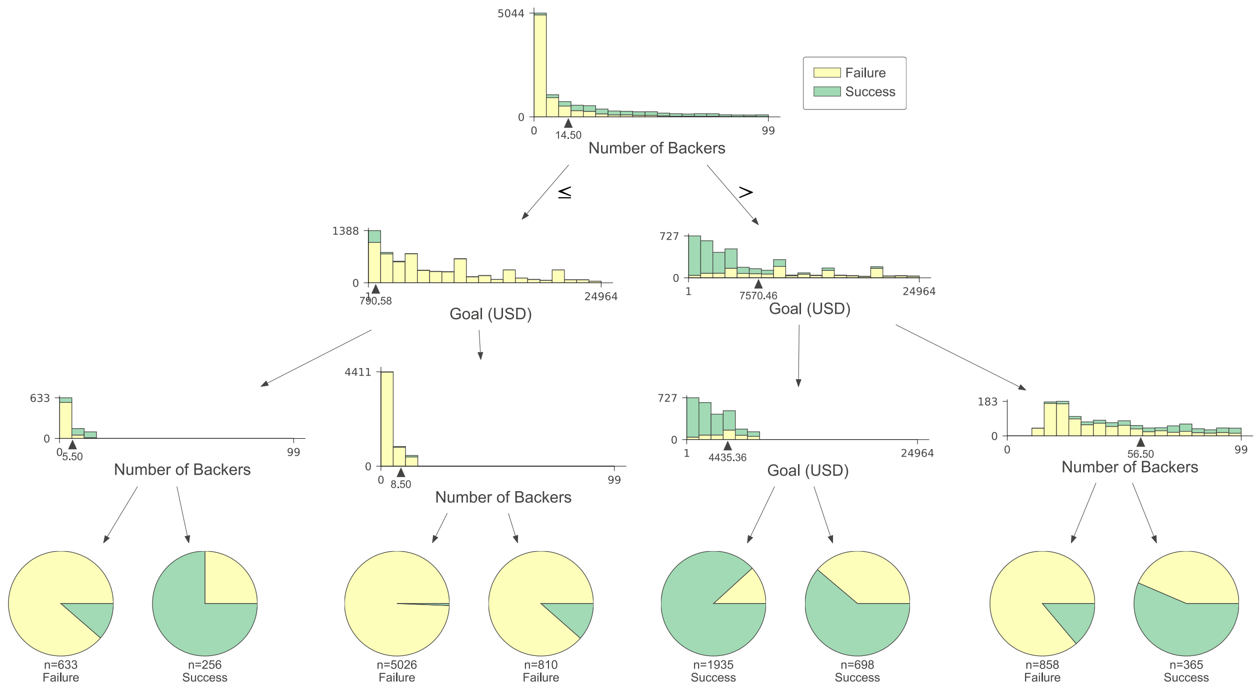


In [402…

```
from dtreeviz.trees import dtreeviz # remember to load the package
viz = dtreeviz(clf, X, y["SuccessfulBool"],
                target_name="",
                   feature_names=["", "Is USA?", "Goal (USD)", "Number of Backer
                   class_names=["Failure", "Success"],)

viz
```

Out[402…




```
<Figure size 3960x1440 with 0 Axes>
```

In [331…

```python
model.summary()
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/discrete/discrete_model.p
y:1810: RuntimeWarning: overflow encountered in exp
  return 1/(1+np.exp(-X))
```

Out[331…

Logit Regression Results

Dep. Variable:	SuccessfulBool	**No. Observations:**	20124
Model:	Logit	**Df Residuals:**	20119
Method:	MLE	**Df Model:**	4
Date:	Fri, 07 Oct 2022	**Pseudo R-squ.:**	0.3795
Time:	19:16:02	**Log-Likelihood:**	-7617.6
converged:	True	**LL-Null:**	-12277.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	-0.3142	0.068	-4.610	0.000	-0.448	-0.181
goal	-6.237e-05	1.64e-06	-38.111	0.000	-6.56e-05	-5.92e-05
backers_count	0.0161	0.000	47.262	0.000	0.015	0.017
launch_to_deadline_days	-0.0181	0.002	-10.000	0.000	-0.022	-0.015
isUS	0.0494	0.043	1.145	0.252	-0.035	0.134

Possibly complete quasi-separation: A fraction 0.11 of observations can be

perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

Prediction

In [345…

```
model.predict(
    [
        [1, 25000, 50, 7, 1],
        [1, 25000, 100, 14, 1],
        [1, 25000, 200, 30, 1],
        [1, 25000, 300, 60, 1],
        [1, 25000, 1000, 90, 1],
    ]
)
```

Out[345…

```
array([0.24093951, 0.38438148, 0.69969011, 0.87087644, 0.99999667])
```

In [333…

```
import math
```

In [342…

```
 math.e ** (-6.237e-05)
```

Out[342…

```
0.999937631944968
```

In [335…

```
math.e ** (0.0161)
```

Out[335…

```
1.0162303033554483
```

In [336…

```
1 - math.e ** -0.0181
```

Out[336…

```
0.017937178834293688
```

In [340…

```
math.e ** 0.0494
```

Out[340…

```
1.0506405229091558
```

In [338…

```
df.shape
```

Out[338…

```
(20124, 20)
```

In [248…

```
df.columns
```

Out[248…

```
Index(['goal', 'disable_communication', 'backers_count', 'category',
       'deadline_weekday', 'launched_at_weekday', 'deadline_month',
       'deadline_day', 'deadline_yr', 'deadline_hr', 'launched_at_month',
       'launched_at_day', 'launched_at_yr', 'launched_at_hr',
       'launch_to_deadline_days', 'SuccessfulBool', 'LaunchedTuesday',
       'DeadlineWeekend', 'isUS', 'pledge_per_backer'],
      dtype='object')
```

In [270…

```
baseline = ["goal", "backers_count", "category", "launch_to_deadline_days", "Suc
```

In [272…

```
df[baseline].describe()
```

Out[272…

	goal	backers_count	launch_to_deadline_days	SuccessfulBool	isUS
count	2.012400e+04	20124.000000	20124.000000	20124.000000	20124.000000
mean	8.806323e+04	185.865335	34.617472	0.299046	0.312512
std	1.299946e+06	1235.778801	11.836983	0.457851	0.463529
min	7.022768e-01	0.000000	1.000000	0.000000	0.000000
25%	4.000000e+03	2.000000	30.000000	0.000000	0.000000
50%	1.348892e+04	12.000000	30.000000	0.000000	0.000000
75%	4.500000e+04	64.000000	40.000000	1.000000	1.000000
max	1.000000e+08	105857.000000	91.000000	1.000000	1.000000

In [285…

```
df["SuccessfulBool"].value_counts()
```

Out[285…

```
0    14106
1     6018
Name: SuccessfulBool, dtype: int64
```

In [288…

```
df[df["SuccessfulBool"] == 1]["backers_count"].describe()
```

Out[288…

```
count      6018.000000
mean        553.332669
std        2200.792317
min           1.000000
25%          39.000000
50%         105.000000
75%         380.000000
max      105857.000000
Name: backers_count, dtype: float64
```

In [296…

```
df["isUS"].value_counts(normalize=True)
```

Out[296…

```
0    0.687488
1    0.312512
Name: isUS, dtype: float64
```

In [ ]: