Kickstarter Project

Group 28 : Denise Mooren, Seeun Park, Steven Emmink, Sungmin Kim

1

Derived features

Features derived from other columns in the dataset

Derived Features

reached_deadline

- Derived from columns ‘created_at’ and ‘deadline’
- Value is 1 (true) if the deadline was reached on time
- Value is 0 (false) if deadline was not met

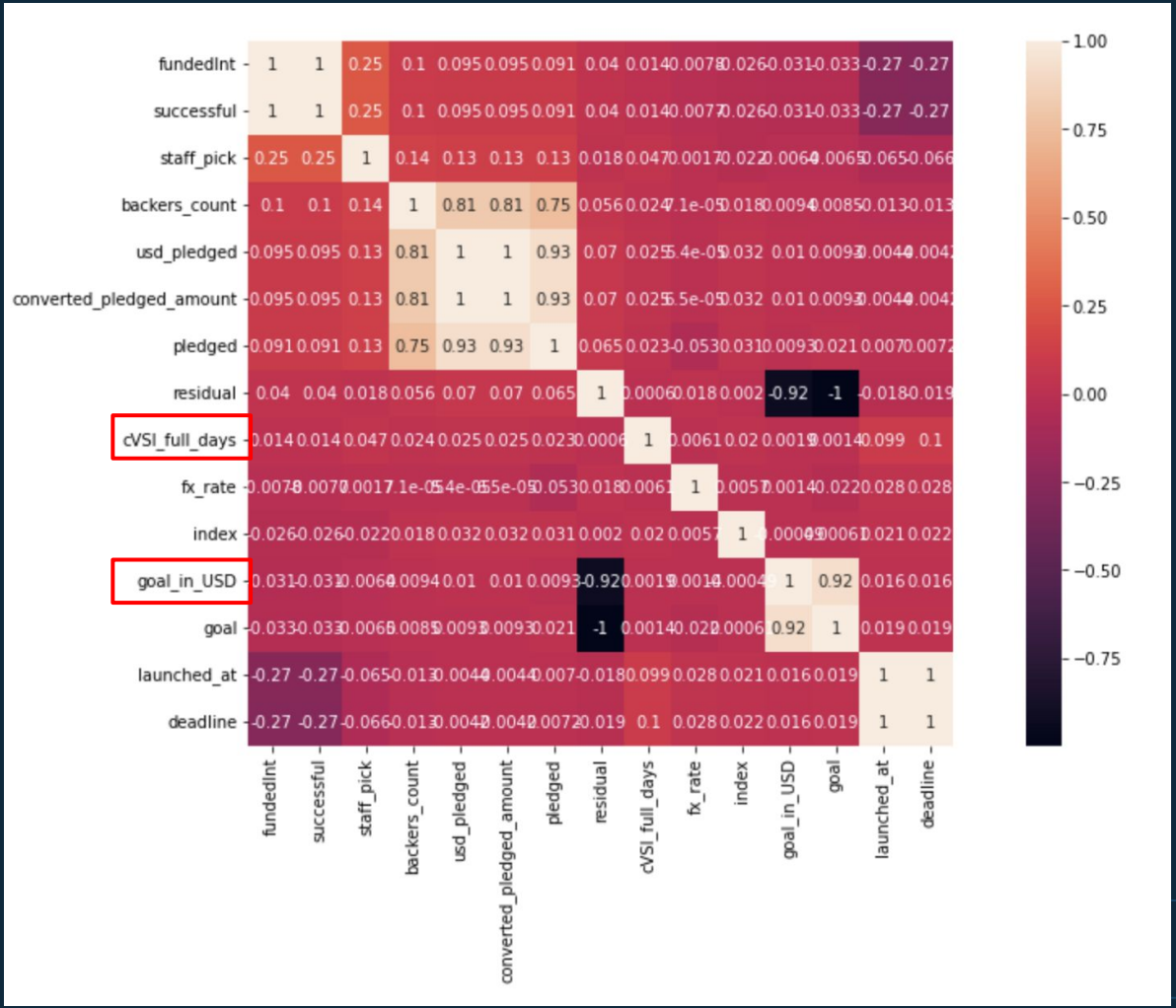
goal_in_USD

- Derived from columns ‘fx_rate’ and ‘goal’
- Goal in USD dollars in floats

cVSl_full_days

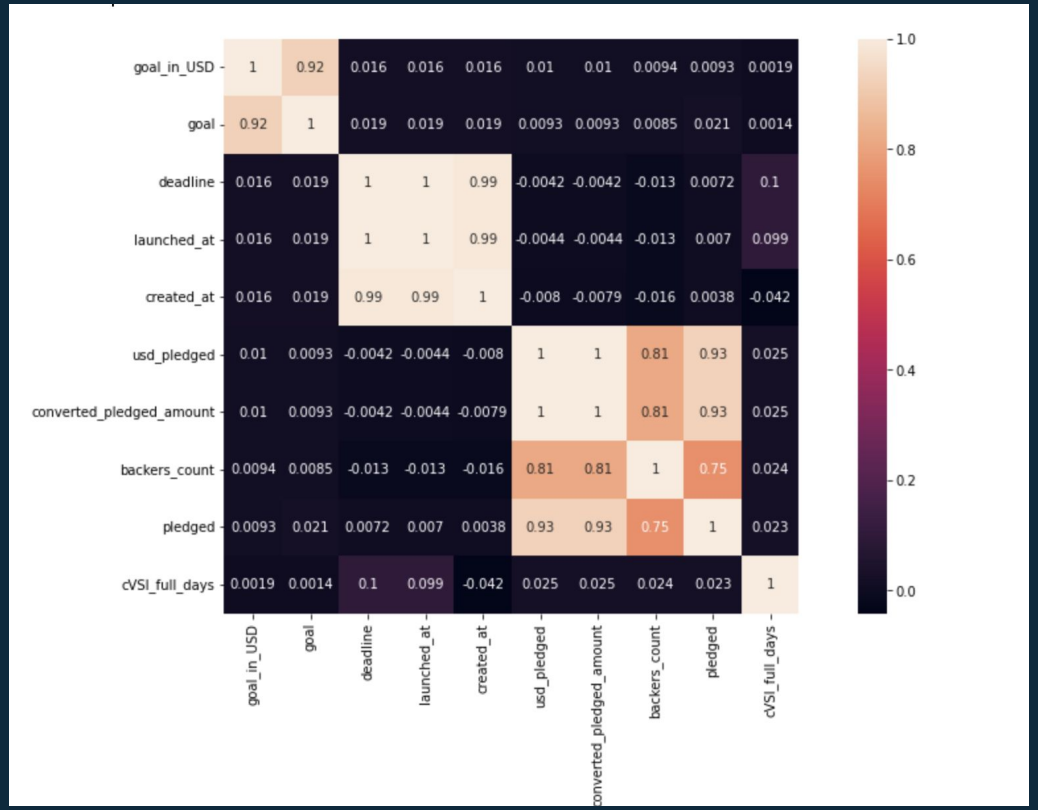
- Derived from columns ‘created_at’ and ‘launched_at’
- Number of days between created and launched date in integers

Heatmap of 15 most correlated features on ‘fundedInt’



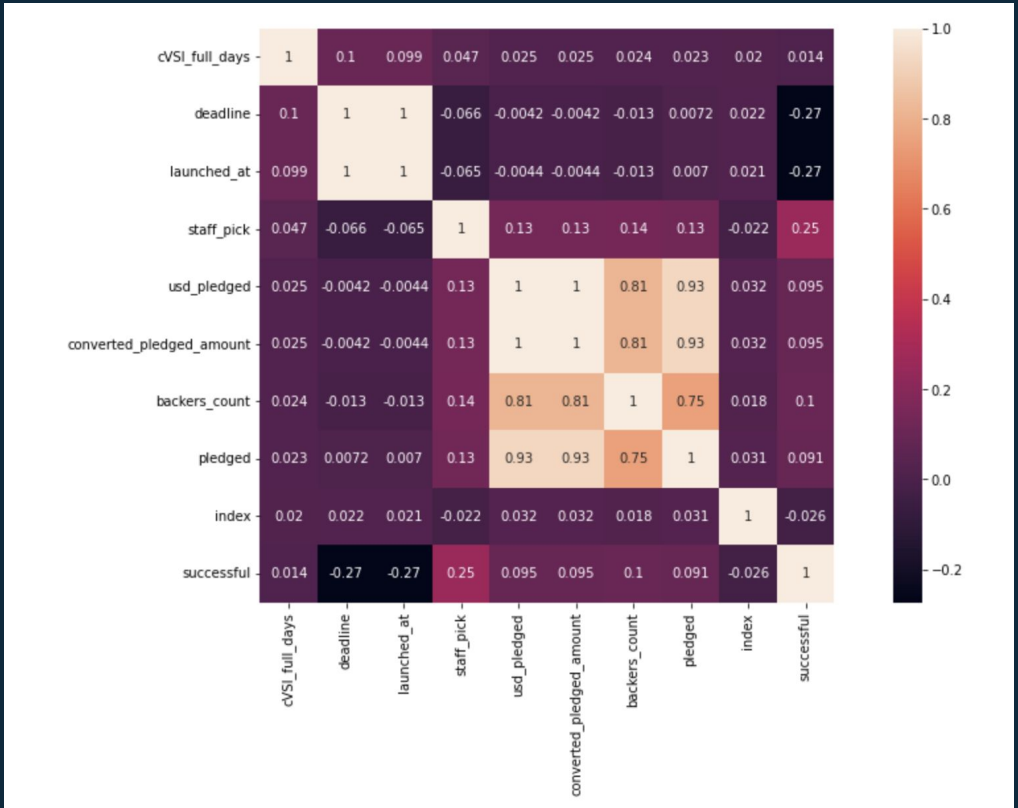
Plots of Derived Features

goal_in_USD



Plots of Derived Features

cVSl_full_days



2

Model Choice & Cross Validation

Design choices, feature selection, and cross validation scheme

Random Forest

- Good at handling categorical and numerical data
- Less sensitive to overfitting

- Features of importance were one hot encoded
  - ‘name’, ‘blurb’, ‘location’ and URL features excluded

- Used SelectFromModel to select most important features for training
  - 20 features selected

Cross-Validation Scheme

Nr of Folds	Accuracy and Variance CV
k = 2	0.74 (+- 0.01)
k = 3	0.74 (+- 0.02)
k = 4	0.74 (+- 0.03)
k = 5	0.74 (+- 0.05)
k = 6	0.75 (+- 0.06)
k = 7	0.73 (+- 0.06)
k = 8	0.74 (+- 0.09)
k = 9	0.72 (+- 0.11)
k = 10	0.72 (+- 0.08)

- Tried several numbers of folds (2 - 10)
- The most optimum number of folds RF: 6

3

Parameters

Hyperparameter tuning for the Random Forest Classifier

Hyperparameter tuning

- Hyperparameters used:

  { 'bootstrap': [True],

  'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],

  'max_features': ['auto', 'log2', None],

  'min_samples_leaf': [1,2,4],

  'min_samples_split': [2, 5, 10],

  'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000] }

- Initially used RandomizedSearchCV
  - Performed 6 different random searches with 3 folds CV

- With the results from the random searches performed a more directed grid search

- Best values: (max_depth=20, min_samples_leaf=4, min_samples_split=3, n_estimators=400)

4

Interpretation

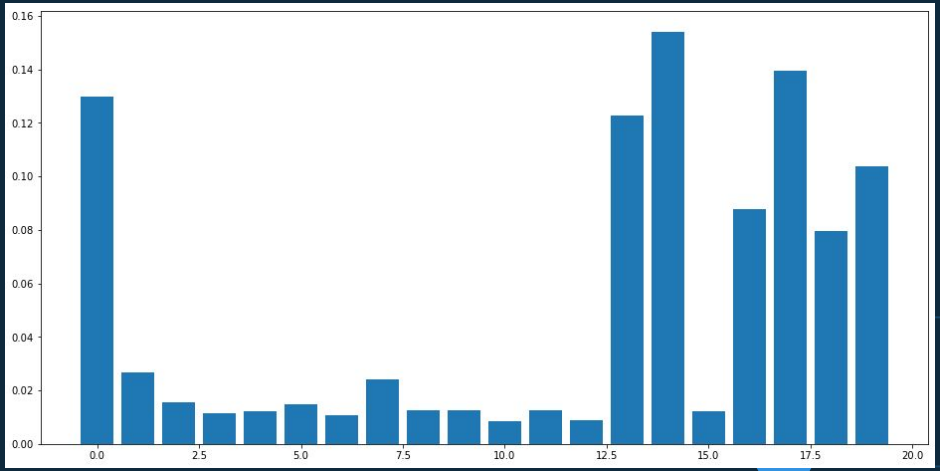
Interpretation of models and feature importances

Most important features

```
Accuracy score on Cross-Validation: 0.75 (+- 0.06)
[0.70126597 0.76042479 0.78862423 0.73264535 0.72542902 0.77313093]
Model score Random Forest: 0.83
```

eli5.show_weights(clf)

	Weight	Feature
feature: staff_pick importance: 0.12972221972557813	0.1297 ± 0.0272	x0
feature: category_art importance: 0.026544981242290783	0.0265 ± 0.0080	x1
feature: category_food importance: 0.015590110066207202	0.0156 ± 0.0143	x2
feature: category_technology importance: 0.011333903468880285	0.0113 ± 0.0155	x3
feature: subcategory_animation importance: 0.012033171059122216	0.0120 ± 0.0042	x4
feature: subcategory_children's books importance: 0.014923077320404108	0.0149 ± 0.0064	x5
feature: subcategory country & folk importance: 0.010748598403896428	0.0107 ± 0.0056	x6
feature: subcategory_hip-hop importance: 0.024127480434843463	0.0241 ± 0.0054	x7
feature: subcategory_mobile games importance: 0.012546559335984198	0.0125 ± 0.0039	x8
feature: subcategory_nonfiction importance: 0.012576008171669855	0.0126 ± 0.0046	x9
feature: subcategory_product design importance: 0.008562319005149747	0.0086 ± 0.0119	x10
feature: subcategory_video games importance: 0.012552721982460045	0.0126 ± 0.0074	x11
feature: subcategory_webseries importance: 0.008872467629744791	0.0089 ± 0.0036	x12
feature: created_at importance: 0.12285449356019512	0.1229 ± 0.1996	x13
feature: deadline importance: 0.15416622270551078	0.1542 ± 0.2193	x14
feature: fx rate importance: 0.012333098685610515	0.0123 ± 0.0035	x15
feature: goal importance: 0.08786941162842647	0.0879 ± 0.0698	x16
feature: launched_at importance: 0.1394400922953605	0.1394 ± 0.2111	x17
feature: cVSl full days importance: 0.07941217766133578	0.0794 ± 0.0171	x18
feature: goal_in_USD importance: 0.10379088561732973	0.1038 ± 0.0715	x19



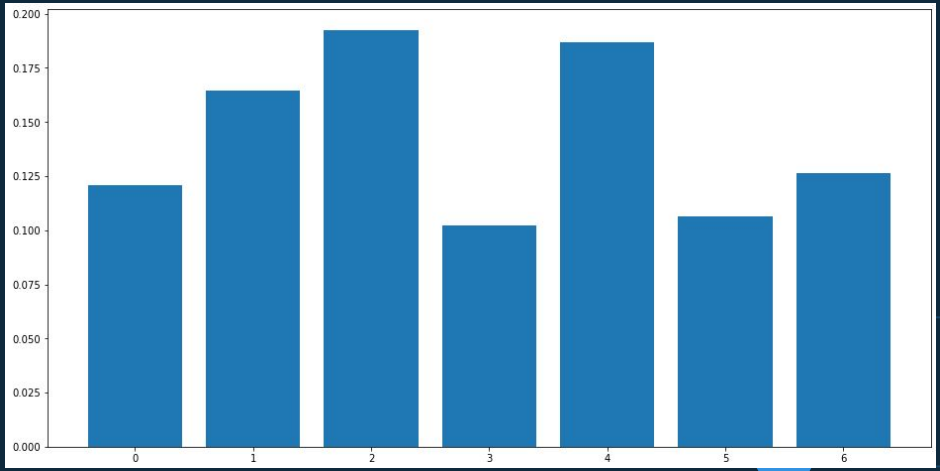
Most important features

```
Accuracy score on Cross-Validation: 0.74 (+- 0.05)
[0.69664607 0.74566509 0.77074459 0.72628547 0.71702868 0.76221049]
Model score Random Forest: 0.85
```

```
feature: staff_pick——importance: 0.12080509261875493
feature: created_at——importance: 0.16465686310589828
feature: deadline———importance: 0.19260615931395197
feature: goal————importance: 0.1021957098335032
feature: launched_at——importance: 0.18693905202184322
feature: cVSl_full_days—importance: 0.10644125314322943
feature: goal_in_USD——importance: 0.1263558699628189
```

```
eli5.show_weights(clf)
```

Weight	Feature
0.1208 ± 0.0210	x0
0.1647 ± 0.1880	x1
0.1926 ± 0.2092	x2
0.1022 ± 0.0630	x3
0.1869 ± 0.2039	x4
0.1064 ± 0.0180	x5
0.1264 ± 0.0608	x6



5

Comparison

Comparison of assigned Naive Bayes Model with Random Forest

Naive Bayes Pros + Cons

STRENGTHS

Gaussian NB = continuous

Categorical NB = categorical

Performs exceptionally well with categorical data

NB

NB

WEAKNESSES

Performance for numerical data is not the best

NB is a poor estimator

Don't have too much faith in predict_proba

```
Model score Gaussain Naive Bayes: 0.61
Model score Categorical Naive Bayes: 0.72
```

Naive Bayes VS Random Forest

```
Accuracy score on Cross-Validation: 0.70 (+- 0.11)
[0.61402088 0.6399064  0.67185672 0.79632796 0.70929709 0.73458735
 0.63522635 0.74835748 0.70947709]
Model score Gaussain Naive Bayes: 0.70
```

```
Accuracy score on Cross-Validation: 0.75 (+- 0.06)
[0.699766   0.76114478 0.78574429 0.73246535 0.72452898 0.77421097]
Model score Random Forest: 0.83
```

Thanks!

Any questions?