Mahmoud Belhaj
Cen Liang
Jonas Cristens

The dataset

1. ID = Internal kickstarter ID
2. Name = Name of the project
3. Main_category = Category of the campaign
4. Category = Sub category of the main_category
5. Currency = Currency used to support the project
6. Deadline = Deadline for crowdsourcing
7. Goal = fundraising goal
8. Launched = When is the project launched
9. Pledge = How much money was gathered by the crowd
10. State = The state of the project (e.g. success, failure)

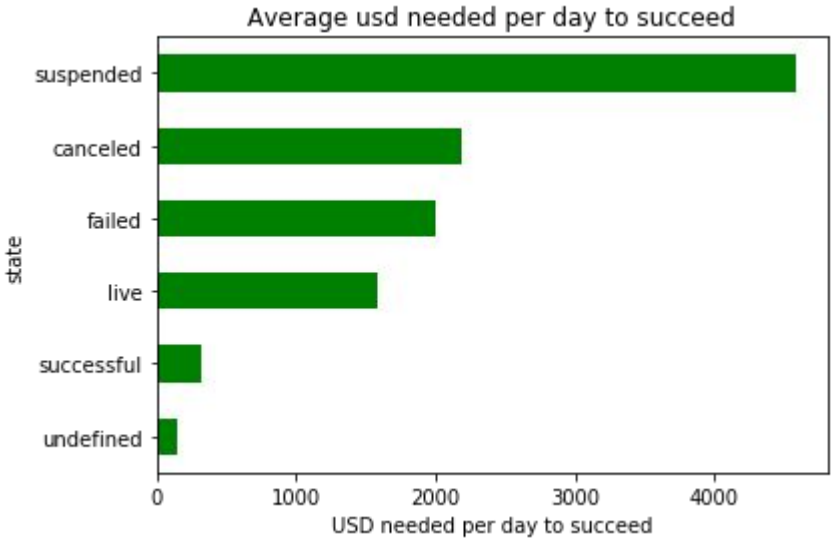
The dataset

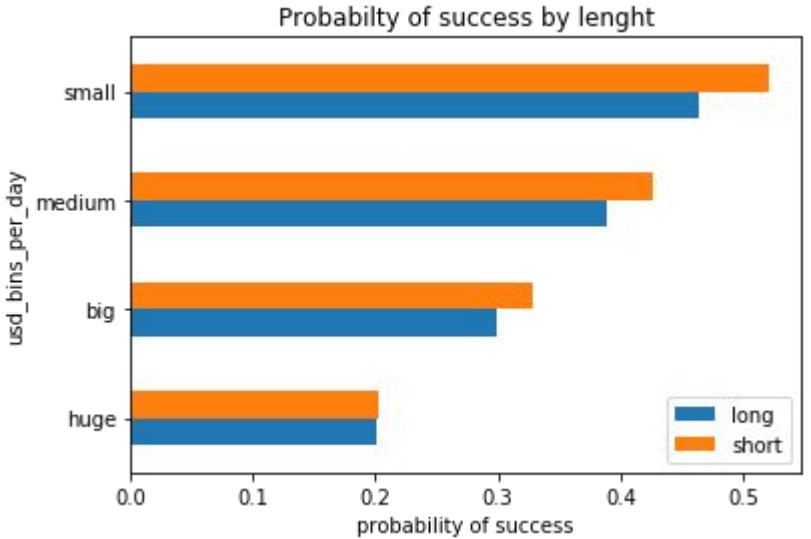
1. Backers = Number of people that backed the project
2. Country = The country of origin
3. Usd_pledged = Pledged amount in USD done by KickStarter
4. Usd_pledged_real = Pledged amount in USD using currency converter
5. Usd_goal_real = Goal amount in USD

Feature engineering

- Number of characters in the name of the project
- Number of words
- Number of syllables
- Launched week, month and weekday
- Days to collect the money
- Average amount needed per day to succeed
- Average amount donated per backer
- Difference between mean goal per category and the goal of the project

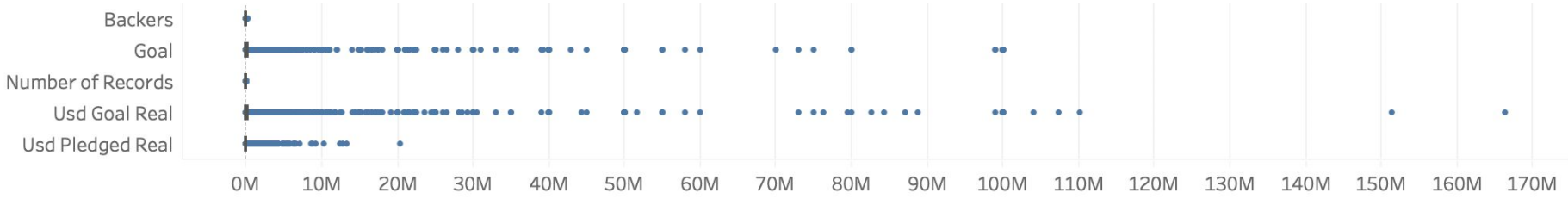
Daily Goal


Average usd needed per day to succeed
suspended
canceled
failed
live
successful
undefined
state
0
1000
2000
3000
4000
USD needed per day to succeed

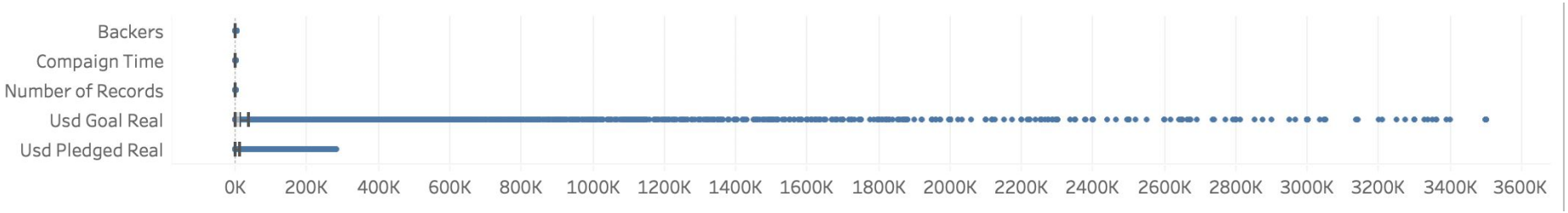

Probabilty of success by lenght
small
medium
big
huge
usd_bins_per_day
long
short
0.0
0.1
0.2
0.3
0.4
0.5
probability of success

Data cleaning

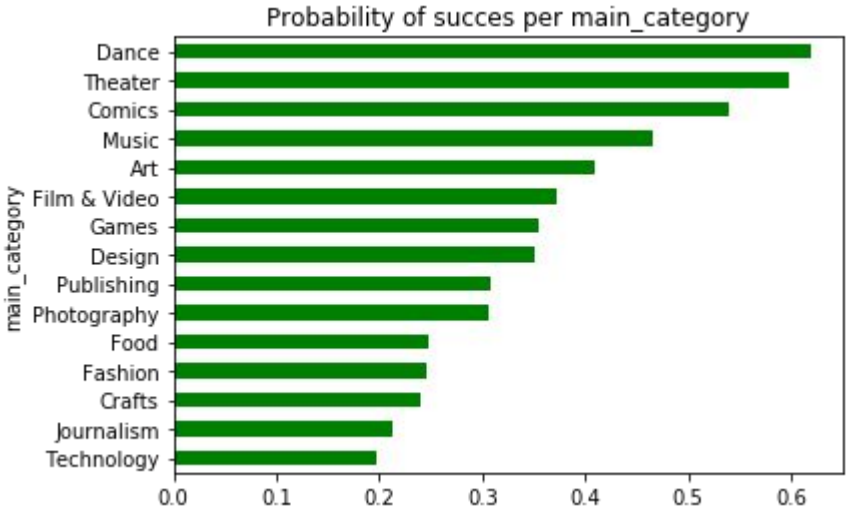
Before

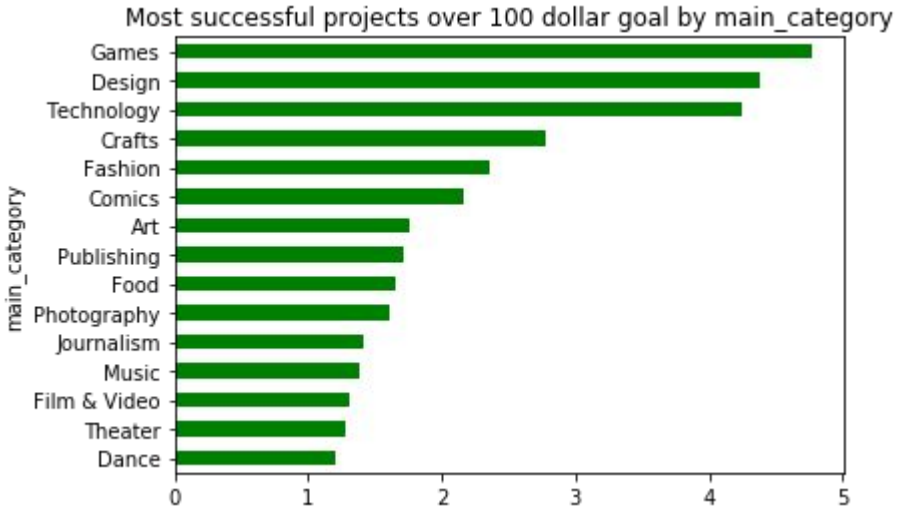


After

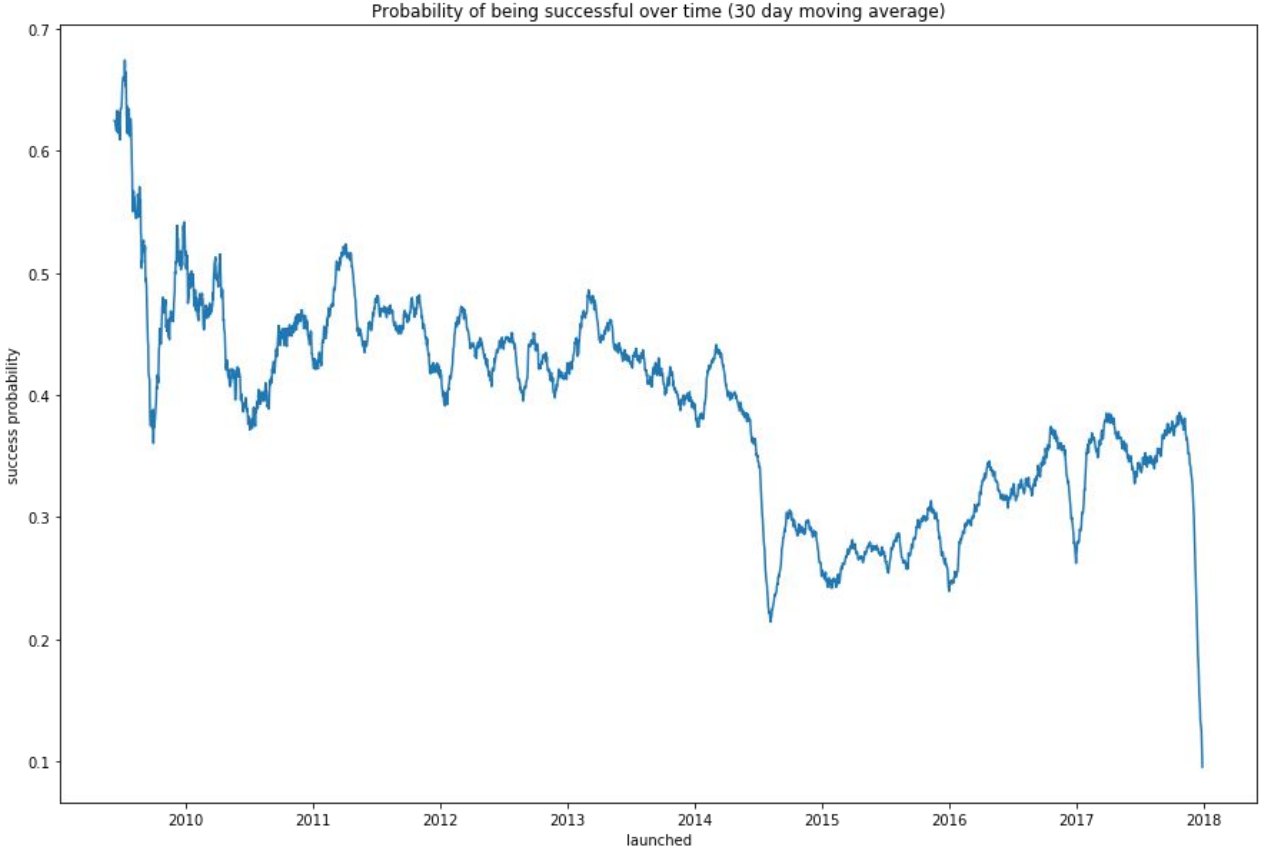


Categories

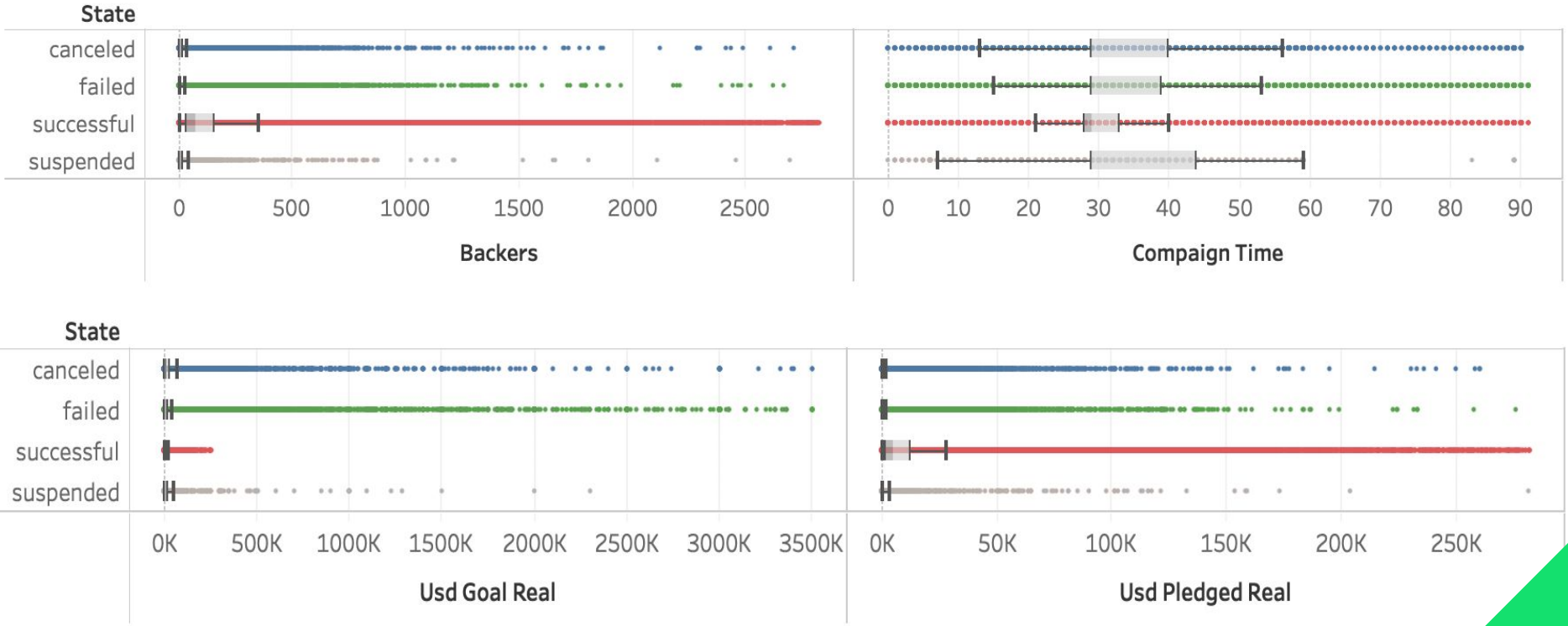

Probability of succes per main_category
main_category
Dance
Theater
Comics
Music
Art
Film & Video
Games
Design
Publishing
Photography
Food
Fashion
Crafts
Journalism
Technology
0.0
0.1
0.2
0.3
0.4
0.5
0.6


Most successful projects over 100 dollar goal by main_category
main_category
Games
Design
Technology
Crafts
Fashion
Comics
Art
Publishing
Food
Photography
Journalism
Music
Film & Video
Theater
Dance
0
1
2
3
4
5

EDA


Probability of being successful over time (30 day moving average)
success probability
launched
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2010
2011
2012
2013
2014
2015
2016
2017
2018

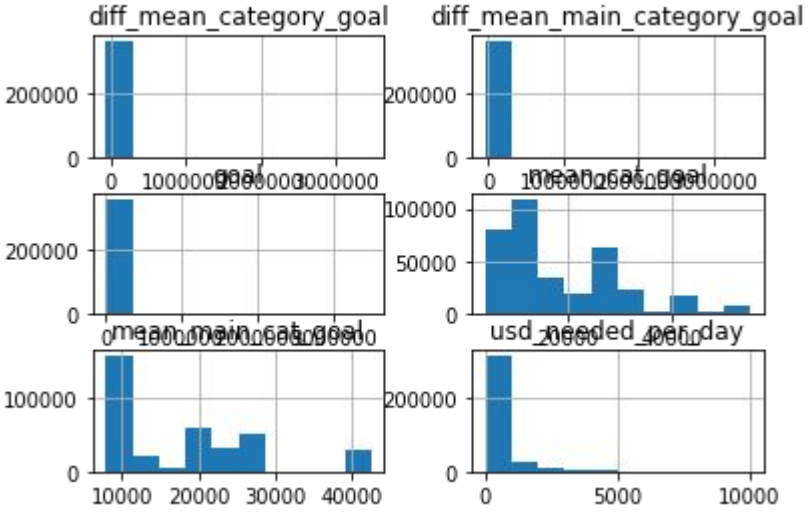
Successful v.s. Others



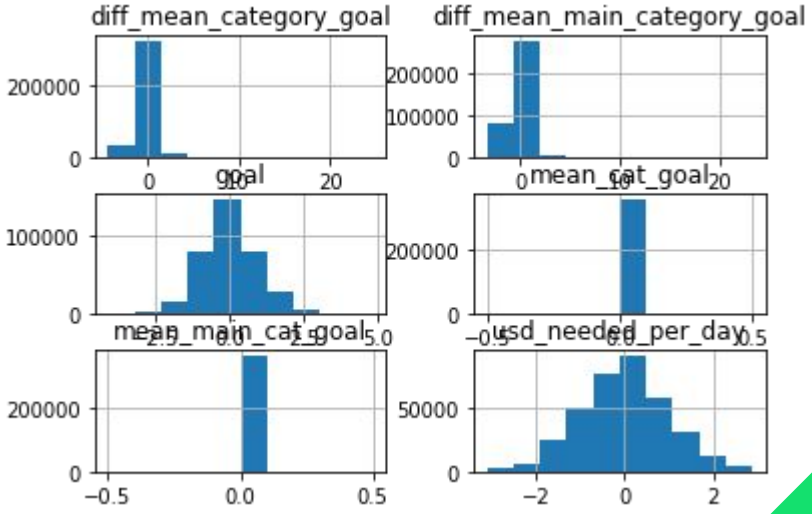
Standardization of the data

```
trans = PowerTransformer().fit_transform(X = df[['goal', 'usd_needed_per_day'\
                                                  , 'mean_cat_goal', 'mean_main_cat_goal'\
                                                  , 'diff_mean_category_goal', 'diff_mean_main_category_goal']])
```

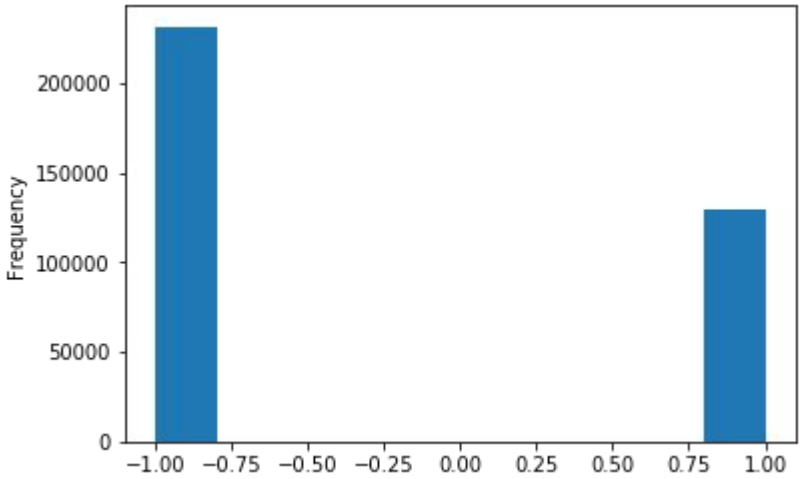
Before

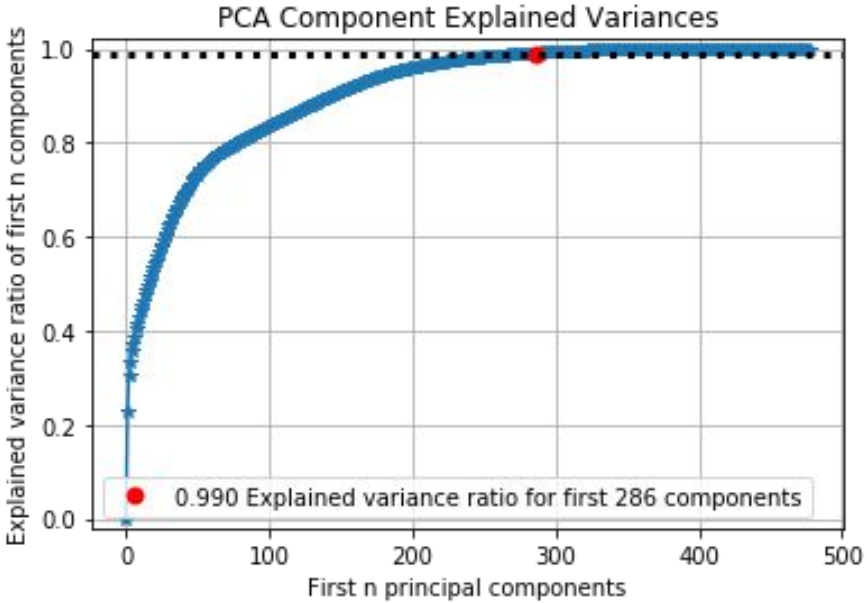


After



Oversampling the data & PCA





Experiments

Model	Accuracy
Logistic regression	69.2%
Extra Tree Classifier	68%
SGDClassifier	68.7%
RBFSampler	66.5%
Neural Network	69%
Gradient Boosting Classifier	69%
XGBoost	Takes too long

Result: LogisticRegression

```
clf = LogisticRegression(C=0.01, n_jobs=-1)
```

	precision	recall	f1-score	support
-1	0.76	0.75	0.76	76354
1	0.57	0.58	0.58	42826
micro avg	0.69	0.69	0.69	119180
macro avg	0.67	0.67	0.67	119180
weighted avg	0.69	0.69	0.69	119180