Predicting Excitment at donorschoose.org OutSystems challenge

Jorge Gomes

jorgemcgomes@gmail.com | https://jcgomes.pt

Context

DonorsChoose.org is an online charity that makes it easy to help students in need through school donations.

Teachers in K-12 schools propose projects requesting materials to enhance the education of their students.

Problem

Help DonorsChoose.org by predicting which projects are likely to be “exciting”.

“**Exciting**” is a business construct meaning that the project will be successful in the platform.

- Fully funded
- have at least one donor referred by a teacher
- have a high percentage of donors leaving an original message
- have at least one donation made with the desired payment means
- have donations from certain desirable donors

DonorsChoose provides the data for all the projects in the platform until 2014.

The objective is to predict the **outcome of future projects**.

Data

projects.csv: various structured information about each project, filled by the teachers when they submit the projects

resources.csv: structured information about the resources requested for each project

essays.csv: project text posted by the teachers (unstructured)

outcomes.csv: information about the outcomes of projects in the training set

donations.csv: information about the donations to each project in the training set

Total of **664098** projects in the provided data. **35** features in the main dataset (projects.csv)

Approach

1. **Load**, **clean**, and **visualise** the projects' data
2. **Prepare** the data for training a model
3. **Train** the model and **assess** its performance
4. Extract **additional features** from the projects' data (feature engineering)
5. Train and assess the model with these additional features
6. Extract more features from **new data sources**
7. Train and assess the model again
8. Evaluate **alternative models**

Technologies

R, RStudio

- \+ data.table, ggplot2
- For data analysis, cleanup, plotting, transformation, etc.
- R Markdown for creating the notebook

Python, PyCharm

- pandas, numpy, sklearn, tensorflow.keras
- For model building and assessment

Cleaning the data

```
$school_state
feature
    CA     NY     NC     IL     TX     FL     SC     IN     GA     OK     PA     TN     MI     LA     MO     VA
126242  73182  43478  40167  39661  30605  18615  17299  15403  14853  14379  14079  12330  12180  12097  10716
    NJ     AZ     MD     MA     UT     NV     OH     CT     WI     CO     MS     DC     OR     AR     AL     KY
 10411   9837   9555   9403   9304   8844   7813   7728   7027   7021   6930   6918   6610   5770   5650   4541
[illegible] [illegible] [illegible] [illegible] [illegible] [illegible]     RI     ID     DE     NE     NH     AK     SD     MT     VT     ND
  3413   3186   2829   2649   2586   2334   2127   2030   1605   1542   1491   1383    990    819    555    483
    La
     3

$school_zip
[1] "Categorical. Unique values:16623"

$school_metro
feature
   urban suburban     <NA>    rural
  349703   152234    81908    80253
```

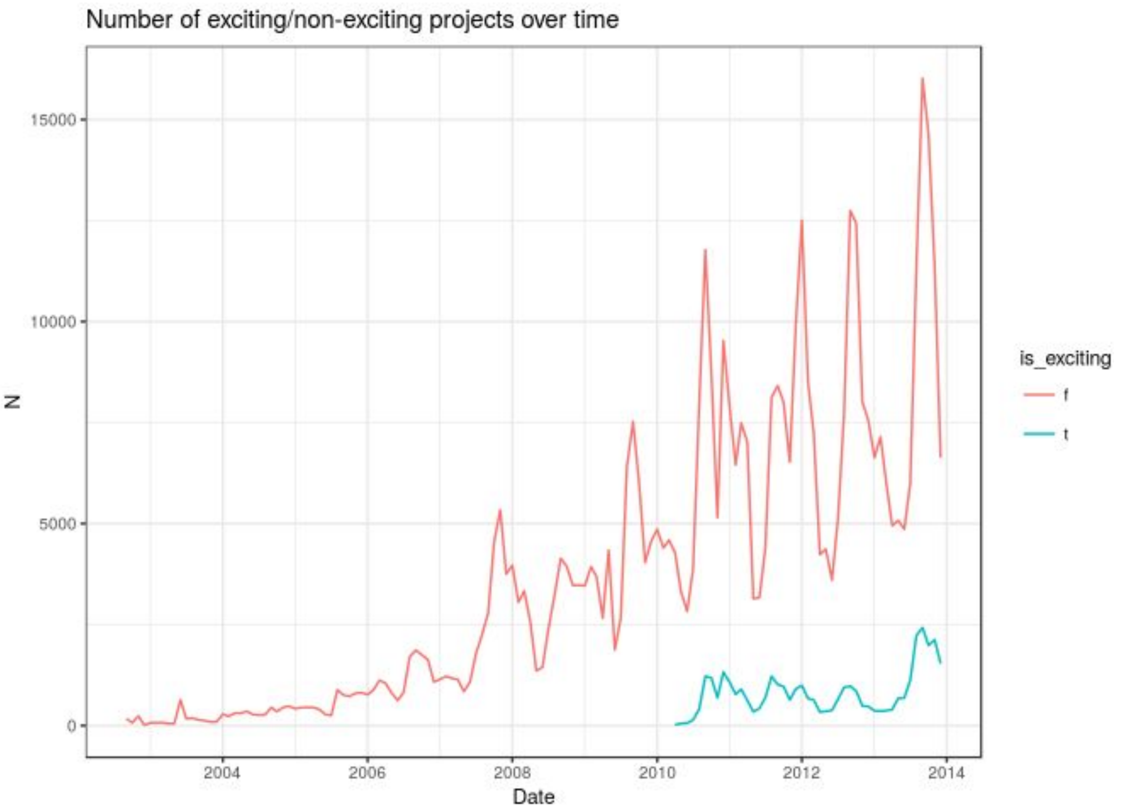
1. Remove the attributes that are **irrelevant** (ID-like features, categorical variables with a non-manageable number of levels)
2. Impute **missing data**:
    a. create a new level for categorical variables with lots of NAs
    b. impute the most frequent level when the number of NAs is negligible
    c. impute the median for numerical variables

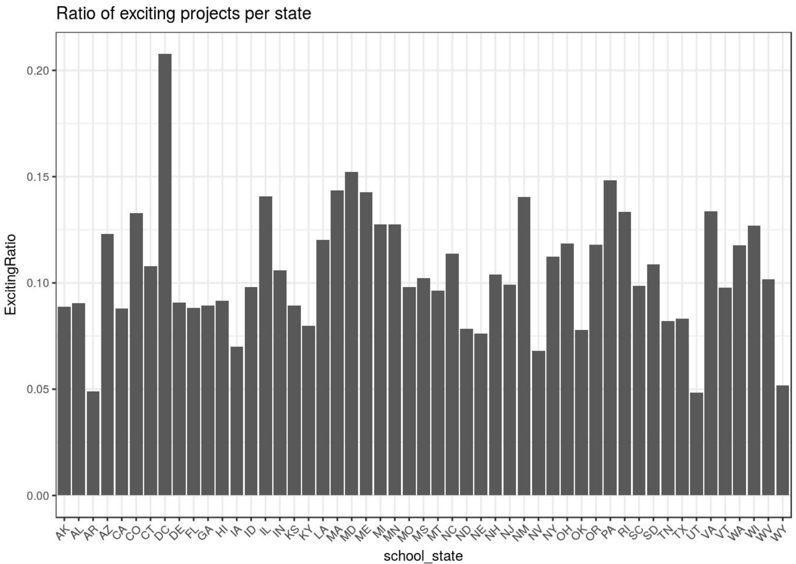
Understanding the data

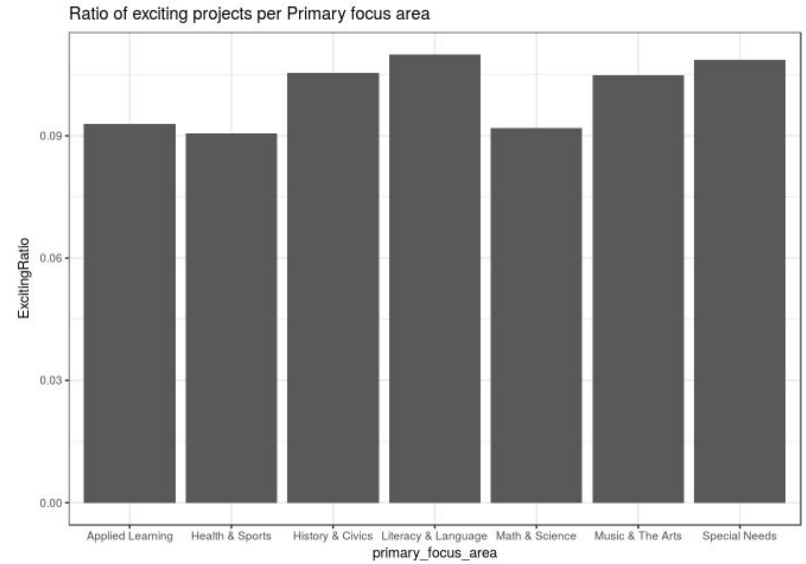
The projects only started to get exciting after 2010

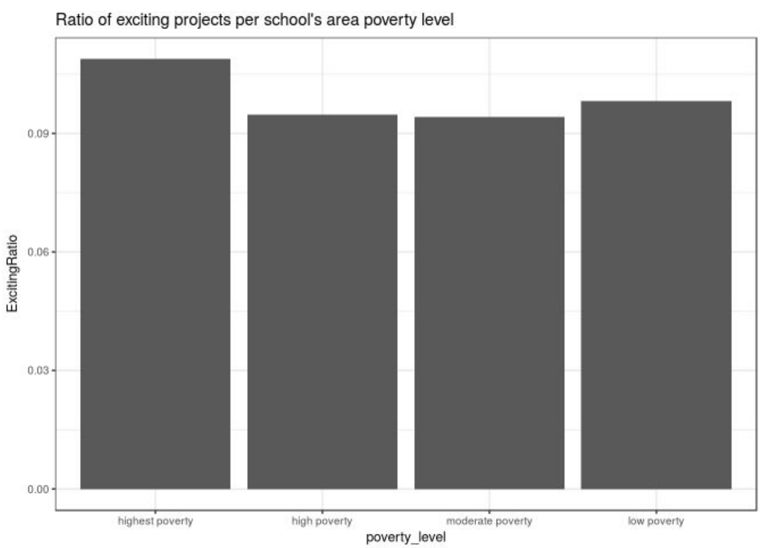
All the data before 2011 is therefore **discarded** for training

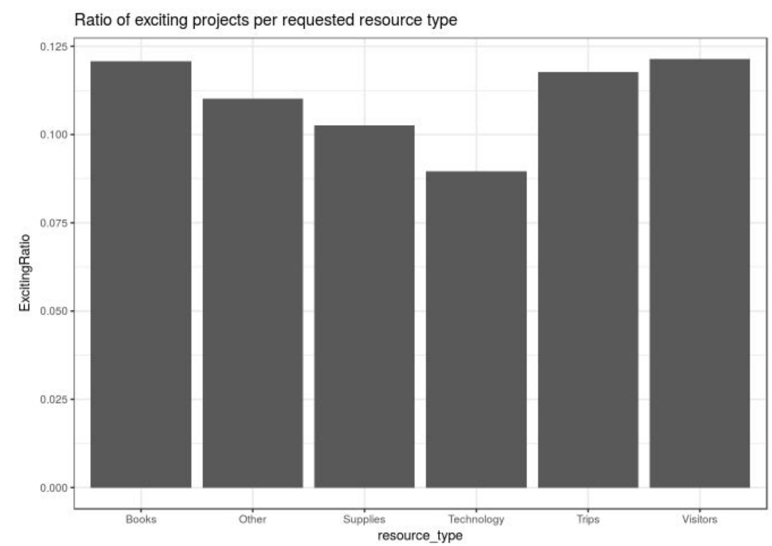
The classes in the dataset are largely imbalanced (6:1)











Looking at intuitively key features for understanding the data

Preparation of the training data

One-Hot-Encoding: create groups of dummy variables for each categorical feature

Split into training and test data

- The data used for testing is the 20% most recent data

Standardise all variables: all features with zero mean and unit variance

The test data will never be seen by the model training, and is standardised with the coefficients calculated based on the training data only

Model assessment

Large class imbalance in the data (6:1). Use metrics that are insensitive to to class imbalance:

Confusion matrix. with a discrimination threshold of 0.5

True Positive Rate (TPR), **False Positive Rate (FPR)**, **Precision**

ROC curve: TPR vs FPR obtained by sweeping the discrimination threshold

ROC-AUC score. The area under the ROC curve.

- Ranging from 0.5 (random guess) to 1 (perfect predictions)
- The metric used by the Kaggle competition

Model building

sklearn.linear_model.LogisticRegression

class sklearn.linear_model. **LogisticRegression** (*penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=1*) [source]

The target variable is the **is_exciting** binary variable. The model will predict the probability of being exciting. Logistic regression is the go-to method for binary classification problems.

Logistic Regression with **L1** (Lasso) regularization

The dataset as a rather large number of features, and is relatively sparse

Logistic Regression with L1 regularization is capable of performing feature selection, and hence it was the first choice for a base model.

Results before feature engineering

nonzeros / # features: 137/146

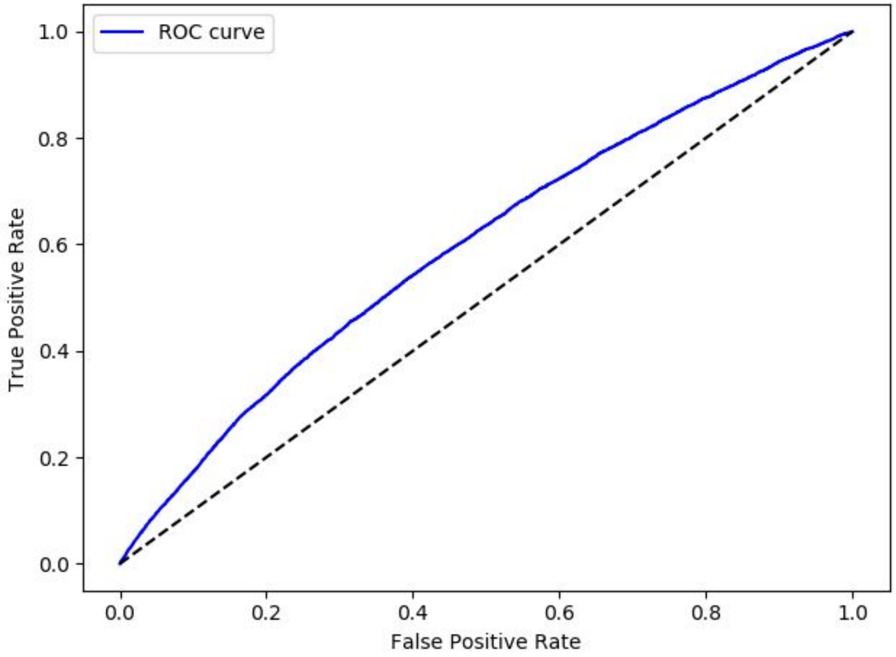
True Positive Rate: 0.4

False Positive Rate: 0.27

Precision: 0.2

ROC AUC score: 0.597

Actual / predicted	non-exciting	exciting
non-exciting	35949	12992
exciting	4838	3222



Feature engineering

Date posted

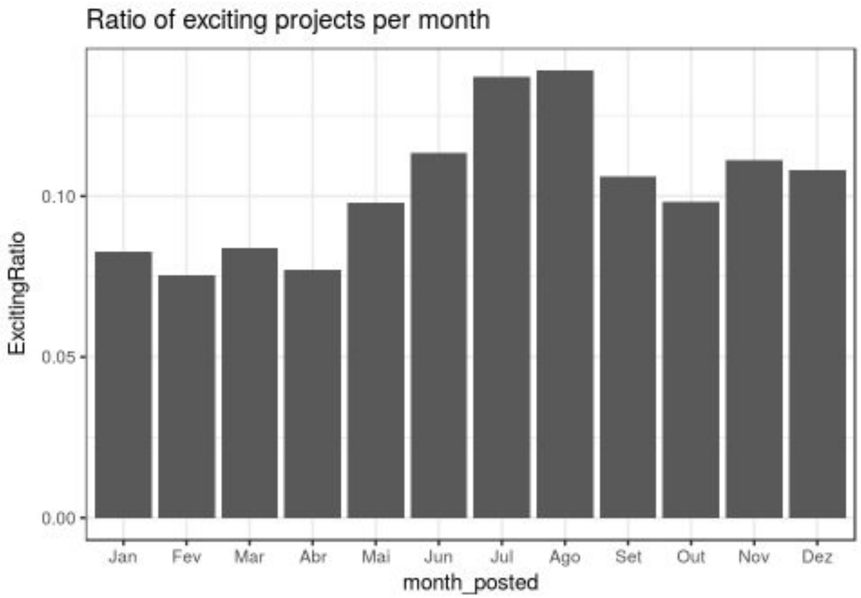
- Year (numeric)
- Month (categorical)
- Weekday (categorical)
- Day of the month (numeric)

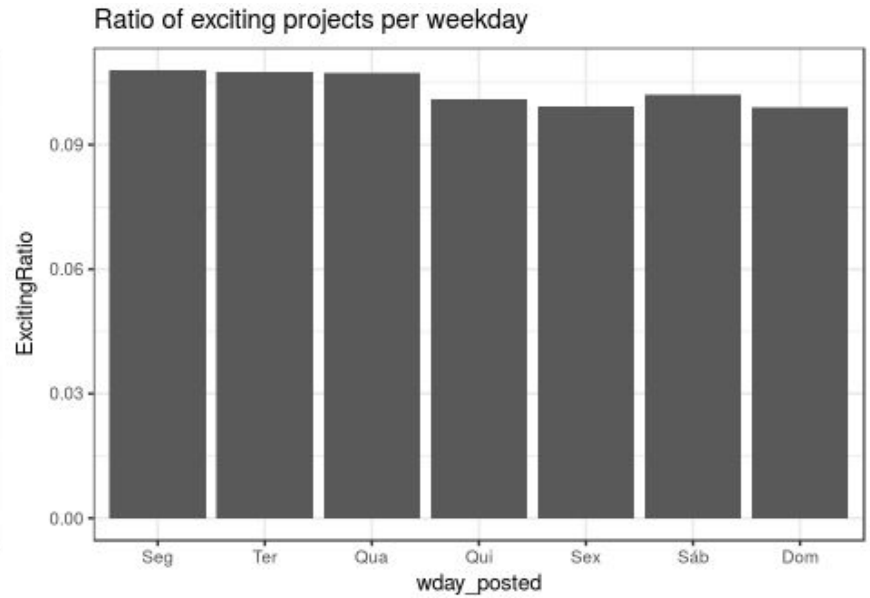
Previous outcomes

- Previous projects submitted by the given school + teacher
- Previous successful projects from the given school + teacher

Resources data

- Average resource price for the given project
- Average quantity of resources for the given project
- Number of different resources requested






```
school_previous_submitted school_previous_success teacher_previous_submitted teacher_previous_success
Min.   :  0.00            Min.   : 0.000          Min.   :  0.000            Min.   : 0.0000
1st Qu.:  2.00            1st Qu.: 0.000          1st Qu.:  0.000            1st Qu.: 0.0000
Median :  8.00            Median : 0.000          Median :  1.000            Median : 0.0000
Mean   : 22.86            Mean   : 1.683          Mean   :  4.226            Mean   : 0.3313
3rd Qu.: 25.00            3rd Qu.: 2.000          3rd Qu.:  3.000            3rd Qu.: 0.0000
Max.   :516.00            Max.   :51.000          Max.   :196.000            Max.   :39.0000
```

```
average_resource_price average_quantity    different_resources
Min.   :    0.15       Min.   :    1.000   Min.   :  1.000
1st Qu.:   19.86       1st Qu.:    1.000   1st Qu.:  1.000
Median :   57.21       Median :    1.226   Median :  3.000
Mean   :  155.10       Mean   :    5.530   Mean   :  5.617
3rd Qu.:  195.79       3rd Qu.:    3.500   3rd Qu.:  6.000
Max.   :93425.78       Max.   :10500.000   Max.   :294.000
NA's   :2873           NA's   :2872
```

Re-training with the new features

nonzeros / # features = 164/172

True Positive Rate: 0.4 → **0.38**

False Positive Rate: 0.27 → **0.22**

Precision: 0.2 → **0.22**

ROC AUC score: 0.597 → **0.621**

Actual / predicted	non-exciting	exciting
non-exciting	35949 → **37993**	12992 → **10948**
exciting	4838 → **4996**	3222 → **3064**

Population data

The project excitement ratio varies largely between states, suggesting that socio-economic factors might play an important role

Each project has the school's **ZIP code**, which indicates its geographical area

Extract publicly available data from the www.irs.gov.
https://www.irs.gov/statistics/soi-tax-stats-individual-income-tax-statistics-zip-code-data-soi

Individual Income Tax ZIP Code Data

ZIP Code data show selected income and tax items classified by State, ZIP Code, and size of adjusted gross income. Data are based on individual income tax returns filed with the IRS and are available for Tax Years 1998, 2001, and 2004 through 2015. The data include items, such as:

- Number of returns, which approximates the number of households
- Number of personal exemptions, which approximates the population
- Adjusted gross income
- Wages and salaries

Population data per zip code

- Number of **households**
- Total **population** number
- Total number of **dependents**
- Fraction of dependents: number of dependents / population number
- Relative distribution of the population over the 6 defined **income classes**

```
    STATE              zipcode          households          population          dependents
 Length:27783       Min.   :    0    Min.   :       80   Min.   :       80   Min.   :        0
 Class :character   1st Qu.:27040    1st Qu.:      580   1st Qu.:     1150   1st Qu.:      340
 Mode  :character   Median :48879    Median :     1920   Median :     3790   Median :     1160
                    Mean   :48878    Mean   :    10579   Mean   :    20595   Mean   :     6846
                    3rd Qu.:70606    3rd Qu.:     7740   3rd Qu.:    14830   3rd Qu.:     4630
                    Max.   :99999    Max.   :17470510   Max.   :35759770   Max.   :12935110
 fraction_dependents       agi1              agi2              agi3              agi4                agi5
 Min.   :0.0000      Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
 1st Qu.:0.2750      1st Qu.:0.3165   1st Qu.:0.2205   1st Qu.:0.1277   1st Qu.:0.07527   1st Qu.:0.06796
 Median :0.3074      Median :0.3707   Median :0.2500   Median :0.1454   Median :0.09524   Median :0.10396
 Mean   :0.3115      Mean   :0.3772   Mean   :0.2455   Mean   :0.1451   Mean   :0.09359   Mean   :0.11074
 3rd Qu.:0.3457      3rd Qu.:0.4286   3rd Qu.:0.2745   3rd Qu.:0.1613   3rd Qu.:0.11250   3rd Qu.:0.15141
 Max.   :0.6471      Max.   :1.0000   Max.   :0.5455   Max.   :0.4444   Max.   :0.36364   Max.   :0.66667
```

Re-training with the new features

nonzeros / # features = 170/182

True Positive Rate: 0.38 → **0.39**

False Positive Rate: 0.22 → **0.23**

Precision: 0.22 → **0.22**

ROC AUC score: 0.621 → **0.621**

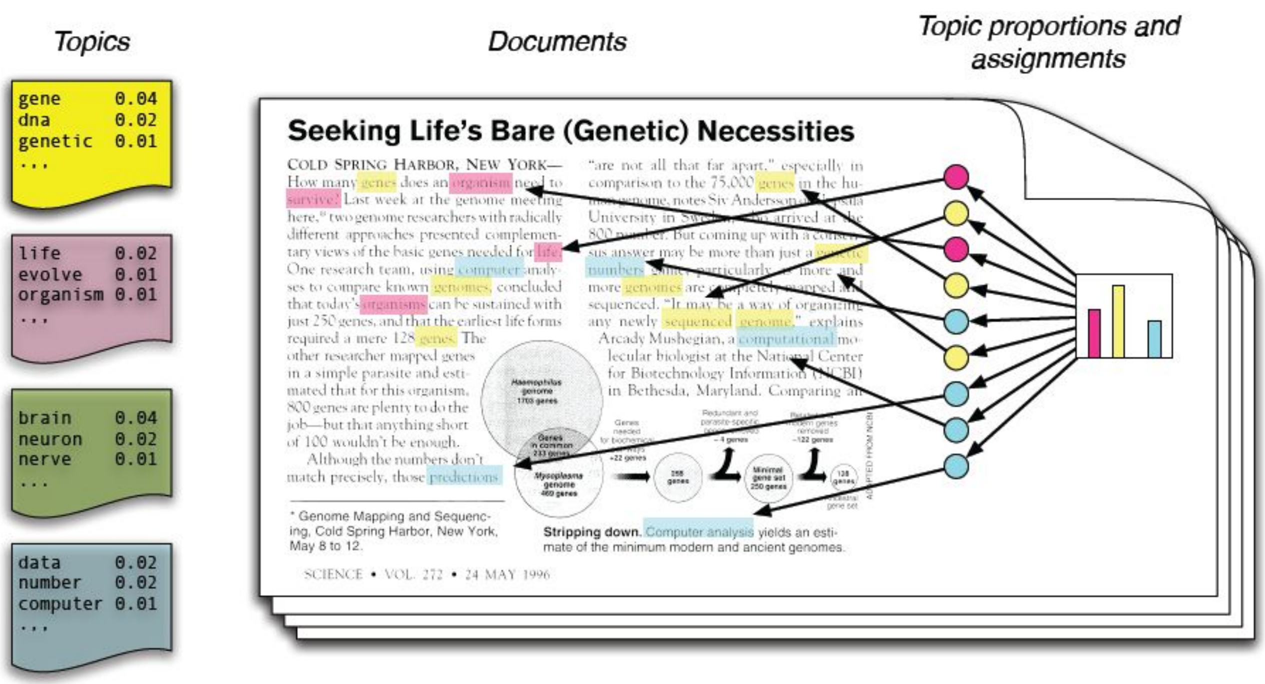
Actual / predicted	non-exciting	exciting
non-exciting	37993 → **37856**	10948 → **11081**
exciting	4996 → **4947**	3064 → **3117**

No significant differences!

Project essays

Use **text mining** to extract meaningful features from the projects' **essays** and **needs statements**

Probabilistic topic models: a text clustering technique that finds non-correlated topics in a given corpus



Text mining workflow

1. Cleanup the essays
    a. Text to lowercase
    b. Remove punctuation and stop words (and, to, with, etc, ...)
    c. Stem the words (going → go, documentation → document, ...)
    d. Remove very frequent and infrequent terms
2. Use topic modelling with a sample of the data to extract the topics (LDA algorithm)
3. Use those topics to calculate the probabilities of each essay belonging to each topic
4. These probabilities become features in the data
5. Do the same for the needs statements

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
"school"	"materi"	"read"	"help"	"high"	"math"	"work"	"class"	"technolog"	"bulli"
"free"	"need"	"level"	"need"	"colleg"	"concept"	"hard"	"time"	"use"	"school"
"lunch"	"resourc"	"reader"	"pleas"	"school"	"problem"	"can"	"one"	"ipad"	"posit"
"receiv"	"help"	"comprehens"	"donat"	"test"	"use"	"help"	"day"	"classroom"	"other"
"reduc"	"provid"	"becom"	"thank"	"prepar"	"understand"	"togeth"	"get"	"app"	"feel"
"titl"	"request"	"struggl"	"support"	"futur"	"manipul"	"get"	"week"	"access"	"help"

Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
"school"	"languag"	"grade"	"make"	"come"	"learn"	"world"	"day"	"can"	"want"
"high"	"english"	"grader"	"can"	"mani"	"lesson"	"live"	"everi"	"ask"	"can"
"mani"	"learner"	"first"	"differ"	"low"	"engag"	"life"	"classroom"	"just"	"feel"
"poverti"	"learn"	"level"	"help"	"school"	"classroom"	"chang"	"daili"	"even"	"like"
"live"	"speak"	"teach"	"mani"	"background"	"interact"	"open"	"come"	"know"	"know"
"citi"	"second"	"second"	"way"	"abl"	"teach"	"see"	"face"	"one"	"abl"

Topic 21	Topic 22	Topic 23	Topic 24	Topic 25	Topic 26	Topic 27	Topic 28	Topic 29
"write"	"school"	"special"	"scienc"	"book"	"school"	"home"	"children"	"activ"
"word"	"fund"	"need"	"experi"	"read"	"communiti"	"mani"	"kindergarten"	"physic"
"use"	"budget"	"educ"	"lab"	"librari"	"program"	"parent"	"learn"	"equip"
"writer"	"money"	"communic"	"handson"	"love"	"build"	"famili"	"child"	"get"
"becom"	"cut"	"disabl"	"explor"	"interest"	"part"	"come"	"young"	"ball"
"stori"	"due"	"skill"	"kit"	"reader"	"particip"	"school"	"letter"	"move"

Topic 30	Topic 31	Topic 32	Topic 33	Topic 34	Topic 35	Topic 36	Topic 37	Topic 38
"camera"	"studi"	"literatur"	"kid"	"opportun"	"learn"	"music"	"year"	"suppli"
"see"	"cultur"	"novel"	"get"	"provid"	"activ"	"play"	"school"	"paper"
"video"	"world"	"class"	"love"	"give"	"fun"	"instrument"	"last"	"pencil"
"pictur"	"histori"	"interest"	"just"	"experi"	"game"	"perform"	"start"	"need"
"share"	"divers"	"set"	"great"	"mani"	"engag"	"school"	"old"	"board"
"document"	"understand"	"discuss"	"dont"	"chanc"	"play"	"program"	"new"	"use"

The most probable terms in the topics extracted for the essays

Re-training with the new features

\# nonzeros / # features = 248/262

True Positive Rate: 0.39 → **0.4**

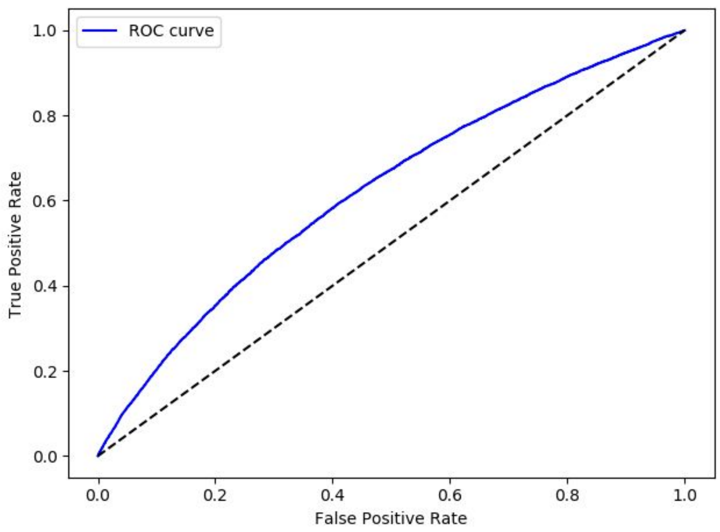
False Positive Rate: 0.23 → **0.23**

Precision: 0.22 → **0.22**

ROC AUC score: 0.621 → **0.623**

Actual / predicted	non-exciting	exciting
non-exciting	37856 → **37463**	11081 → **11478**
exciting	4947 → **4837**	3117 → **3223**

No significant differences!



Kaggle submission

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
kaggle.csv	a few seconds ago	1 seconds	2 seconds	0.61361

Complete

Jump to your position on the leaderboard

46	▲ 18	shzu-IntCMP-ML01	0.61565	109	4y
47	▲ 23	AntonK	0.61544	52	4y
48	▲ 40	yoyow110w	0.61410	24	4y
49	▲ 1	BookEx.org	0.61313	7	4y
50	▲ 21	Catalyst@UOM	0.61197	82	4y

49th place out of 472 participants

The ROC-AUC score achieved in the challenge's test data is very close to my own test data

Final remarks

Extremely challenging problem. Even the winning models on Kaggle achieved a poor/fair performance.

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲1	'STRAYA			0.67813	213	4y
2	▼1	DataRobot			0.67319	220	4y
3	▲22	ChaoticExperiments (KIRAN R)			0.67297	69	4y

My best model had a relatively good **true positive rate** (0.4), but low **precision** (0.22), and **ROC-AUC** score of **0.623**.

Only one in five projects predicted as exciting would actually become exciting.

Model is unreliable, and would have limited uses in a business setting.

Future directions

We know the different criteria that are used to consider a project exciting or not. Alternative approach:

1. Predict the probability of fulfillment of each of the criteria independently.
2. Calculate the probability of a project being exciting based on those probabilities.

Parallel directions:

- Fine tune the models and experiment with more models.
- Experiment with feature selection and understand the importance of each feature.
- This would also provide important information for the end user.

Other non-reported experiments

Logistic regression with a Stochastic Gradient Descent Optimiser

- Same results, faster to train!

Logistic regression with L2 regularisation

- Worse results

Logistic regression with ElasticNet regularisation (L1+L2)

- Same as L1 regularisation

Deep Neural Networks with Tensorflow/Keras

- Significantly worse results
- Too many hyper-parameters to experiment with, too little time