DONOR'S CHOICE

Capstone 2

Bill Murphy

8/12/18

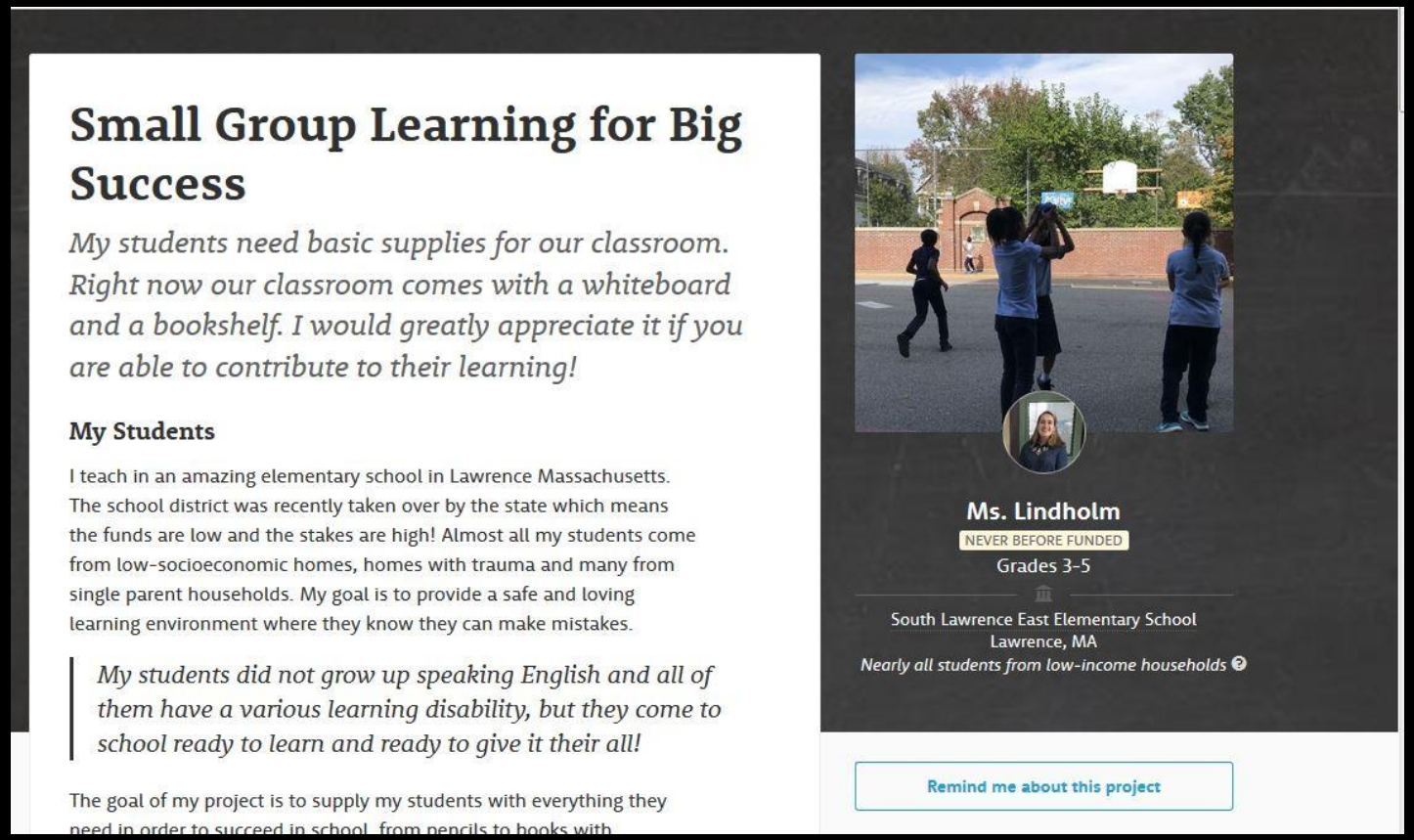
WHO IS DONOR'S CHOICE?

- ***Support a classroom. Build a future.***

- *Teachers all over the U.S. need your help to bring their classroom dreams to life. Choose a project that inspires you and give any amount.*

- *"Our team of 80 has vetted and fulfilled over 600,000 classroom project requests that range from butterfly cocoons, to robotics kits, to Little House on the Prairie. Many of us are former teachers, so our operation feels like a cross between a startup and a schoolhouse."*

WHAT PROBLEM NEEDS TO BE SOLVED?



- 186,000 proposals last year
- Each proposal is currently reviewed by a staff member→ 2,325/staff member per year
- Number of proposals is growing rapidly, and the present system is unsustainable.
- Need proposals that help will reduce the load on staff

WHAT DATA IS AVAILABLE?

kaggle

581 total participants
Top roc_auc score:
0.82812

Trials top 3:
161, 112, 80

- Data source is a previously completed Kaggle competition
- 3 Types of Data Supplied:
  - Numerics: # previous projects, item costs, etc.
  - Categorical: subject, state, teacher title (Mrs., Ms.,Mr., etc)
  - Text Fields: Descriptions of resource needs, essays
- Training Data:
  - 186,000 lines of project requests
  - 85% approval rate as baseline

BASELINE TO JUDGE SUCCESS

- Kaggle used ROC
- Better: Confusion Matrix

Actual

Prediction	0	1
0	0	0
1	27,909	158,151

Actual

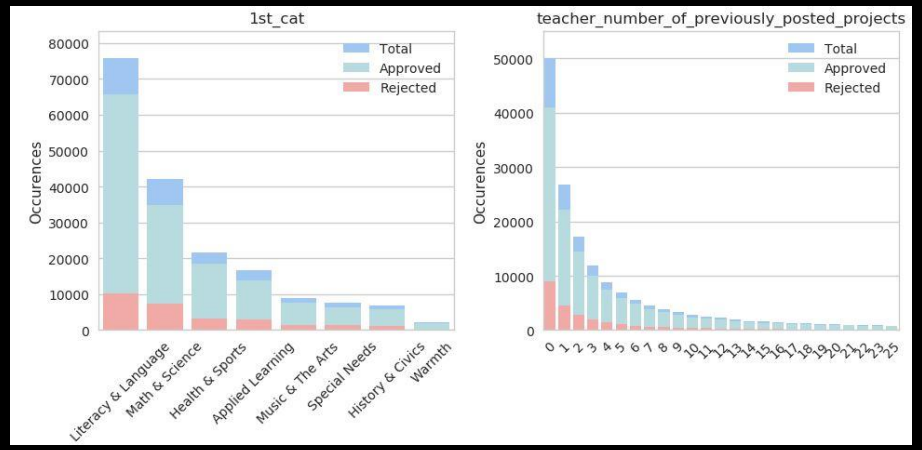
Prediction	0	1
0	0	0
1	15%	85%

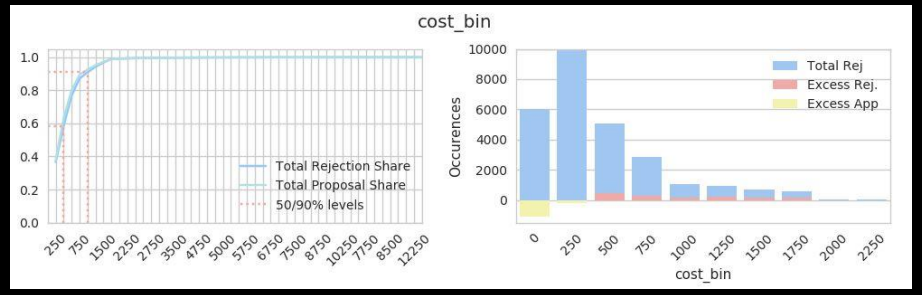
ANALYSIS STRATEGY:

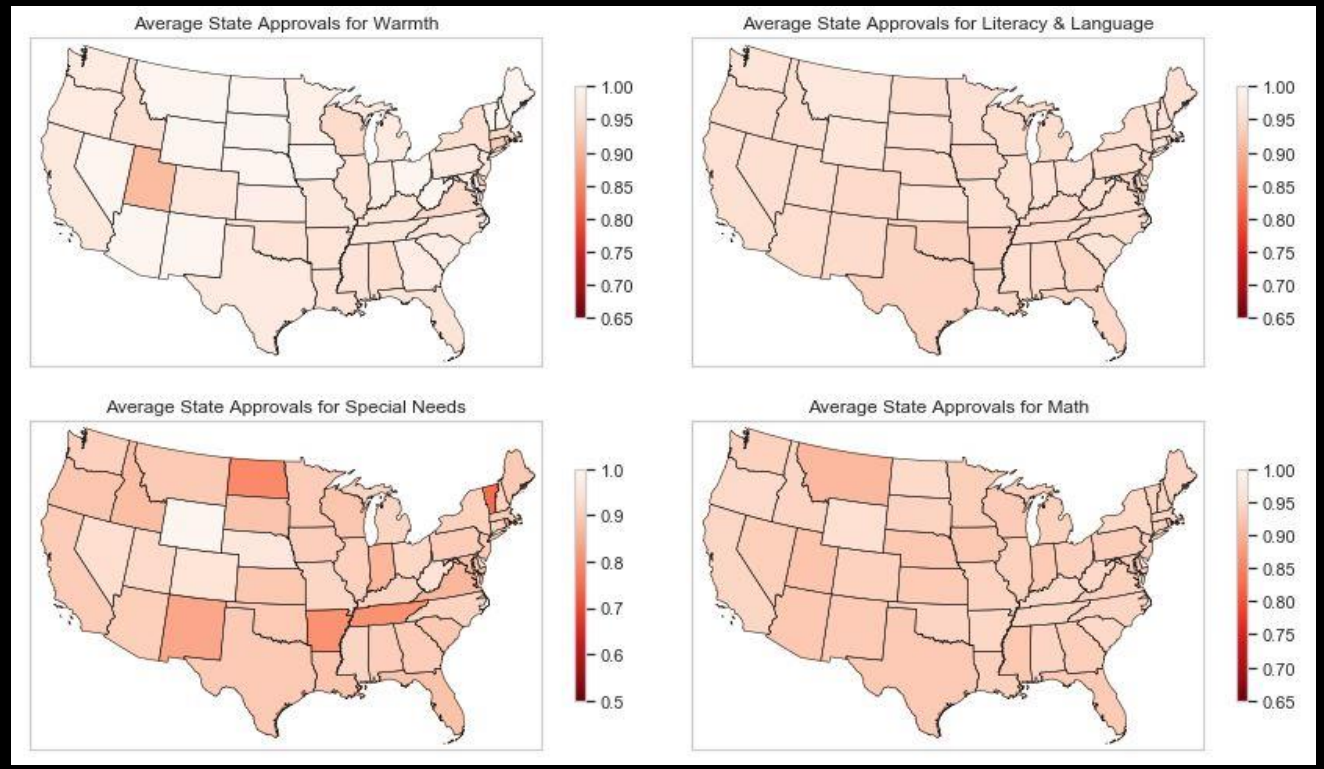
- Clean missing values, Unicode entries, non-English punctuation
- Basic visualizations. analyze baseline
- Added Cost information, re-analyzed
- Added simple text features (spelling, capitalization, etc.) re-analyzed
- Added varying numbers of TF-IDF features, re-analyzed

- Multiple classifiers but limit to more "traditional" classifiers to test differences against winning solution. Used Logistic Regression, Gradient Boost, and Random Forest from sklearn. Added LGB which has a different api.

THE DATA: IT ALL LOOKS LIKE IT HAS THE SAME SHAPE, UNTIL IT DOESN'T…





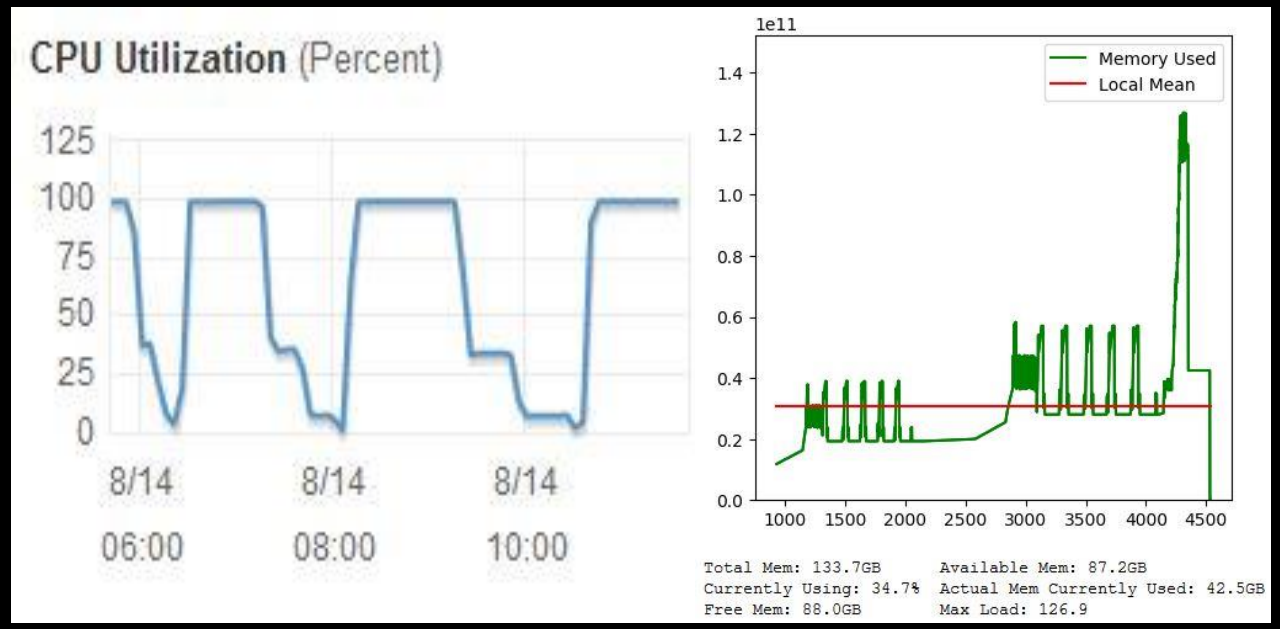


SOME ANALYSIS DETAILS

- Rounded most derived variables on the assumption that humans don't typically think in 64 bit math space.
- Limited textual analysis to ascii and English. There is some room for improvement here if there were more time.
- No feature reduction was included, again due to time constraints.
- Simple word scoring caused leakage, but offers room for improvement with some development. Classifier choice?

THE ARMS RACE IS OVER

AWS EC2 Instance: R54xl
(16 processors, 128gb memory)
Still some failures



My Laptop:
(8 processors, 16BG memory)

0.726050 0.01950 0.131670 0.12278

JUDGING SUCCESS

vs.

- Kaggle used ROC: Best Kaggle ROC Score: 0.76067 winner 0.82812
  Place: 279/581 48th percentile
  Submissions: 20 51/top10, 161 #1

- Improvement in FP rate: 1.8%/15% → 12% improvement

- Confusion Matrix: Before and After

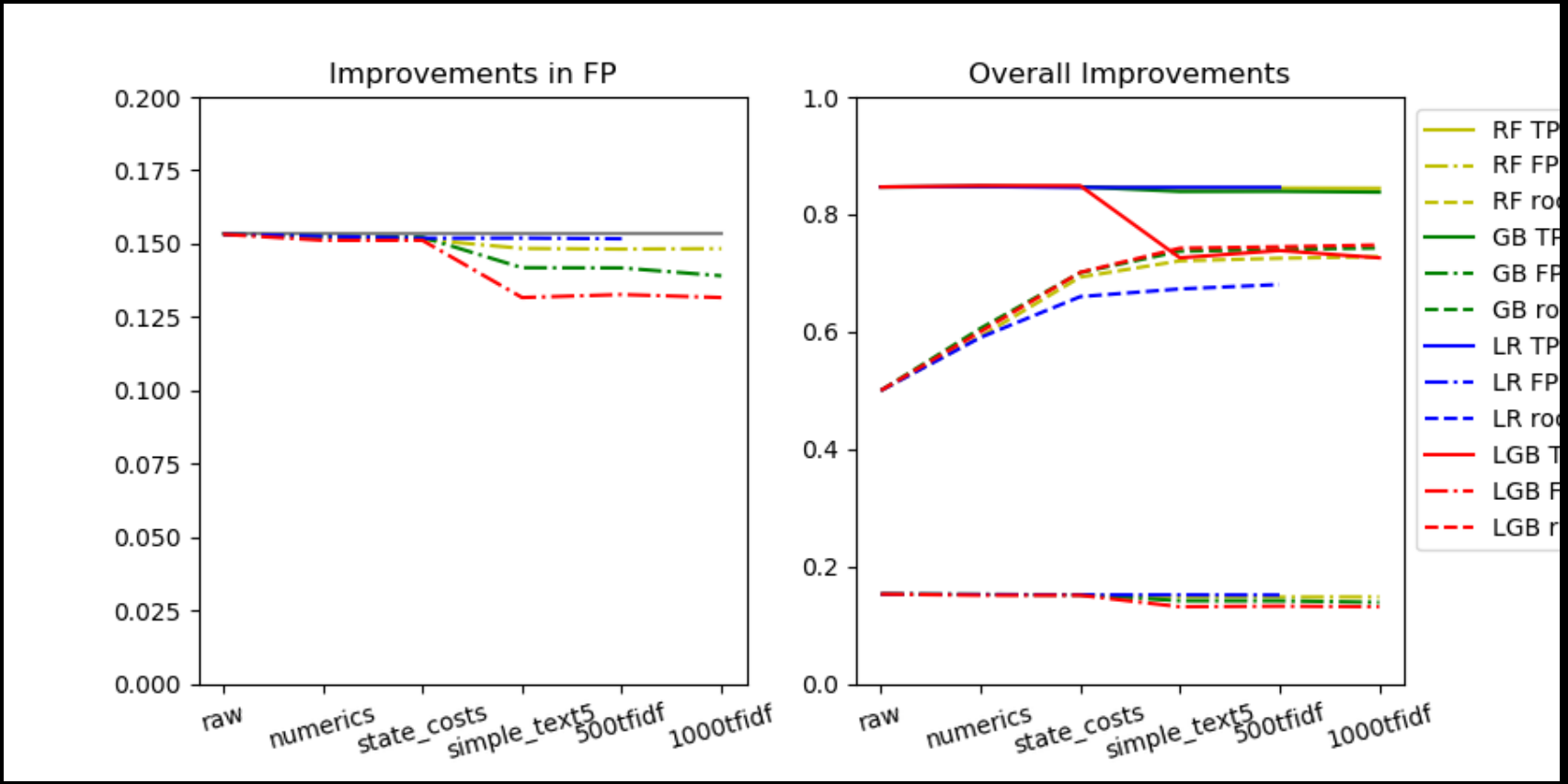
Actual

Prediction	0	1
0	0	0
1	15%	85%

Actual

Prediction	0	1
0	2.0%	12.3%
1	13.2%	72.5%

A PERSPECTIVE ON RESULTS



CLIENT RECOMMENDATIONS

- Before using the models, change your training programs, with an increased focus on Math/Science and Special Needs. High leverage in bringing those rejection ratios to the mean.

- Develop guidelines around the simple text screens. That was the single biggest grouping for improvement in the FP rate.

- Make more geographic segmentation available. There is almost certainly district to district differences that outweigh the state to stat variations.

IF I HAD IT TO DO OVER:

- Different decisions on how to structure text features. Specifically, I would try separating the resources and essays.
- I would use larger AWS instances (or spark.) Some of the classifiers did not complete with more than 1.5K features.
- I would focus on computationally (and memory) efficient classifiers.

SUMMARY:

- PROBLEM: Binary classification, 180K training datapoints, total features (post tfidf) were >5000. Required critical decisions at each step.

- Achieved a 12% improvement in the target parameter. Decent from a data science perspective, but pragmatically, this would not benefit the client proportionally. More work is needed.

- Light Gradient Boost was by far the most effective classifier that I tried due mainly to its efficiency.