SCHOOL BUDGET LINE ITEM CLASSIFICATION

A MULTI-CLASS, MULTI-LABEL CLASSIFICATION PROBLEM

01AUGUST2018

MARK SAUSVILLE
SAUS@IEEE.ORG

THE CUSTOMER

- Education Resource Strategies http://www.erstrategies.org/
- Non-profit consulting organization – helping school districts be more strategic and effective in their spending.
- Work product: Given budget data from a school system or district, classify all spending by line item.
- Analyses produced by human budget analysts (slow, error-prone).
- The analyses allow ERS to understand how schools are spending money and tailor their strategy recommendations to improve outcomes.

THE GOAL

- Machine learning model to allow human analysts to complete work much faster by eliminating time-consuming hand analysis. ERS will be able to serve its client base (public schools and school districts) more quickly and efficiently.
- Eliminate (or mitigate) bottleneck.

THE PROBLEM

Correctly label every budget line item in nine different classifications

Function	Object_Type	Operating_Status	Position_Type	Pre_K	Reporting	Sharing	Student_Type	Use
Teacher Compensation	NO_LABEL	PreK-12 Operating	Teacher	NO_LABEL	School	School Reported	NO_LABEL	Instruction
NO_LABEL	NO_LABEL	Non-Operating	NO_LABEL	NO_LABEL	NO_LABEL	NO_LABEL	NO_LABEL	NO_LABEL
Teacher Compensation	Base Salary/Compensation	PreK-12 Operating	Teacher	Non PreK	School	School Reported	Unspecified	Instruction
Substitute Compensation	Benefits	PreK-12 Operating	Substitute	NO_LABEL	School	School Reported	Unspecified	Instruction
Substitute Compensation	Substitute Compensation	PreK-12 Operating	Teacher	NO_LABEL	School	School Reported	Unspecified	Instruction

THE DATA

- Hosted by DrivenData (www.drivendata.org)
- The competition: Box Plots for Education (live through March 2019)
  - Training Set
    - 400k rows, 9 columns labels, 14 columns text features, 2 columns numerical
  - Holdout Set
    - 50k rows, 14 columns text features, 2 columns numerical
  - The DrivenData tutorial
    - Hosted by DataCamp - **Machine Learning with the Experts: School Budgets**

THE PROJECT

- Acquire the data
- Explore the characteristics of the data
- Analyze the structure and performance of the models from DrivenData tutorial
- Independently produce a competitive model

THE DATA: A CLOSER LOOK (1) – TARGETS

Target Column	Number of labels	Sample of labels
Function	37	'Teacher Compensation', 'NO_LABEL', 'Substitute Compensation'…
Object_Type	11	'NO_LABEL', 'Base Salary/Compensation', 'Benefits'…
Operating_Status	3	'PreK-12 Operating', 'Non-Operating', 'Operating, Not PreK-12'
Position_Type	25	'Teacher', 'NO_LABEL', 'Substitute'…
Pre_K	3	('NO_LABEL', 'Non PreK', 'PreK')
Reporting	3	'School', 'NO_LABEL', 'Non-School'
Sharing	5	'School Reported', 'NO_LABEL', 'School on Central Budgets'…
Student_Type	9	'NO_LABEL', 'Unspecified', 'Special Education'…
Use	8	'Instruction', 'NO_LABEL', 'O&M'…

THE DATA: A CLOSER LOOK (2) – FEATURES

Feature Name	Feature Value
Object_Description'	NaN
'Text_2'	SPECIAL EDUCATION INSTRUCTION
'SubFund_Description'	LOCAL
'Job_Title_Description'	Teacher, Special Education
'Text_3'	NaN
'Text_4'	NaN
'Sub_Object_Description'	NaN
'Location_Description'	NaN
'FTE'	1.0
'Function_Description'	NaN
'Facility_or_Department'	NaN
'Position_Extra'	NaN
'Total'	67397.91883
'Program_Description'	NaN
'Fund_Description'	NaN
'Text_1'	NaN

THE DATA: A CLOSER LOOK (3) – LABEL EXAMPLE

Target	Label
Function	Teacher Compensation
Use	Instruction
Sharing	School Reported
Reporting	School
Student_Type	Special Education
Position_Type	Teacher
Object_Type	Base Salary/Compensation
Pre_K	NO_LABEL
Operating_Status	PreK-12 Operating

THE DATA: A CLOSER LOOK (2) – FEATURES

Feature Name	Feature Value
Object_Description'	NaN
'Text_2'	SPECIAL EDUCATION INSTRUCTION
'SubFund_Description'	LOCAL
'Job_Title_Description'	Teacher, Special Education
'Text_3'	NaN
'Text_4'	NaN
'Sub_Object_Description'	NaN
'Location_Description'	NaN
'FTE'	1.0
'Function_Description'	NaN
'Facility_or_Department'	NaN
'Position_Extra'	NaN
'Total'	67397.91883
'Program_Description'	NaN
'Fund_Description'	NaN
'Text_1'	NaN

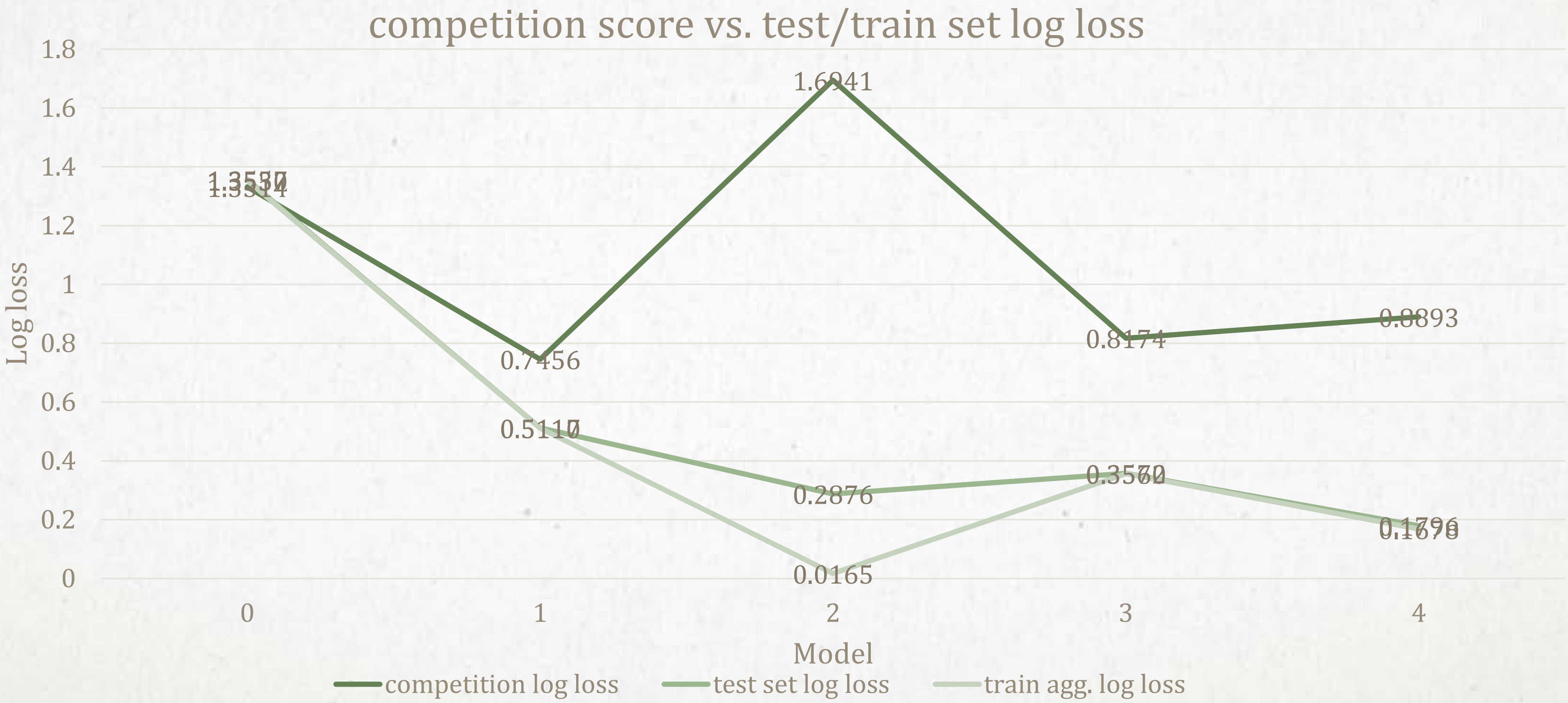
KEY METHODOLOGIES

- One-hot encode labels
  - 9 columns => 104 binary columns
- Merge all text features
  - 14 columns string => 1 column string
- Transform to sparse word-count vectors
  - CountVectorizer/Hashing Vectorizer
- Feature Interaction
  - Select best and compute Cartesian product - all combinations of best features
- OneVsRestClassifier(LogisticRegression())

THE MODELS (DRIVENDATA)

- Mod0
  - Numerical only
- Mod1
  - Text and Numerical data, pipeline, ransformers and CountVectorizer
- Mod2
  - RandomForestClassifier replaces the OneVsRestClassifier demonstrating the flexibility provided by the pipeline.
- Mod3
  - Bigrams
- Mod4
  - HashingVectorizer, SelectKBest, feature interaction

THE RESULTS


competition score vs. test/train set log loss
Log loss
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1.6941
1.3514
0.7456
0.8174
0.8893
0.2876
0.0165
0.1796
0
1
2
3
4
Model
competition log loss
test set log loss
train agg. log loss

THE ISSUES

- Aggregate log loss doesn't provide any insight into the classification. Where is it right? Where is it wrong? Why?
  - Response: Coded tool to go from flat probability predictions to original labels
  - Result: Detailed metrics for each label
    - Accuracy, Precision, Recall, F1, confusion matrix

- Test results improve - scored results on holdout do not improve
  - Difference between training data and holdout set
    - Nothing can be done?

MY MODELS

- DrivenData models skip steps
- My approach:
  - One step at a time so we can separate effects of each tool/technique
- Leverage knowledge gained.

BASIC MODELS

Ignoring numerical features made a big difference.

model	comp. score	agg. log loss	agg. F1 score	comment
mod0	**1.3314**	1.356	0.441	numerical features only
mod0_1		1.323	0.441	same as mod0, but use standard scaler before prediction
mod0_1a		1.295	0.454	scaling + convert total to absolute value
mod0_2		1.362	0.406	same as mod0 but use standard scaler and default imputer before prediction
mod1	**0.7546**	0.512	0.853	pipeline, numerical features and text features (fillna with empty string; combine all text columns within row; default count vectorizer)
mod1_1		0.094	0.974	same as mod1 but ignore numerical data
mod1_1_1	0.6827	0.094	0.974	same as mod1_1; work around n_jobs=-1 bug for faster fit

ADDING BIGRAMS AND FEATURE SELECTION

Bigrams help; we need all the features.

	comp. score	agg. log loss	agg. F1 score	comment
mod3_1	**0.6599**	0.0573	0.982	text features only, add bigrams
mod3_1a	**0.7531**	0.0593	0.982	text features only, add bigrams, add scaler
mod3_2		0.305	0.900	same, but reduce to 300 features using SelectKBest
mod3_2a		0.0798	0.976	same, but reduce to 3000 features using SelectKBest
mod3_2b		0.0589	0.982	same, but reduce to 15000 features using SelectKBest
mod3_3		0.0580	0.982	same, but reduce to 15000 features using SelectFromModel

ADD FEATURE INTERACTION

No help from feature interaction, even keeping all original features

model	comp. score	agg. log loss	agg. F1 score	comment
Mod4	**0.8893**	0.1796	0.866	300 text features with interactions
Mod4_1600	**0.7774**	0.1489	0.978	1600 text features with interactions (slow)
twoC_besties	**0.7373**	0.0565	0.976	200 best features interactions plus all original features
Mod400	**0.7307**	0.0577	0.976	400 best features interactions plus all original features

THE MISSING PIECE

- Great performance on test set(s).
  - Best models have aggregate log loss < 0.05; Aggregate F1 score > 0.98 on test set
- So-so submitted holdout predictions
- Something is missing
  - Can you guess?

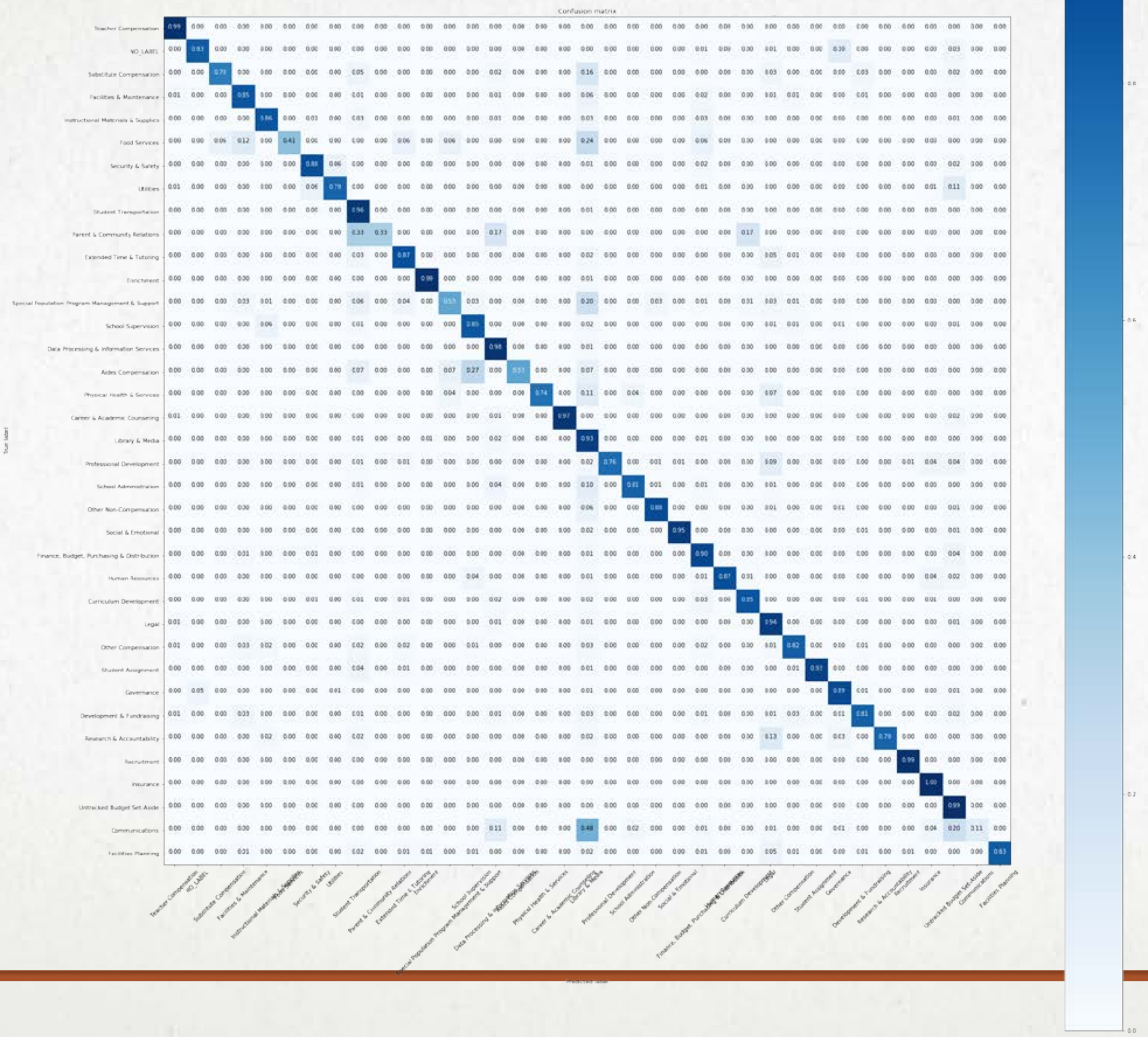
REGULARIZATION

- Our models have great fit to the test data (all of it; many train/test splits)
- Unregularized competition score: 0.6599
- With the best regularization(0.033): 0.5228
- 4^{th} in the competition

RECOMMENDATIONS TO THE CLIENT

- The best scoring classifier we produced has worst case accuracy (and F1 score) of 95% on the test set and is ***sufficient to solve the business problem***.
- Our experience with a range of models tells us that better than 95% average (over all targets) accuracy/F1 can be expected from models with log loss of 0.52 (our best model's score on the holdout set).
- Across all the models, the lowest accuracy was on the target, 'Function'. This target has 37 possible labels, some of which are very rare. Human analysts should pay close attention to this target to validate the model's choice of label.
- To further focus human analysts, the confusion matrices for each target from the classification can be examined to get a better idea which labels are most likely to be labeled incorrectly. The figure below shows a normalized confusion matrix for the target ‘Function’. The darker entries off the diagonal show where significant misclassification has occurred.

EXAMPLE CONFUSION MATRIX



FUTURE WORK

- The models from DrivenData are a sketch, not a roadmap.
- Other classifiers
  - Ensemble models
  - Bagging (to reduce variance)
  - Boosting (XGB)
- Single classifier per target
- Preprocessing for numerical variables
  - If noise can be reduced, may increase predictive power.