Subreddit classification from unidentified post text and metadata

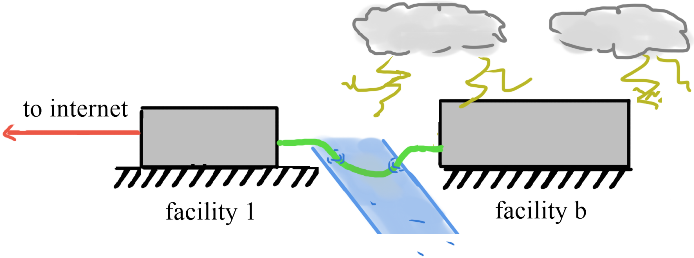
David Tersegno
DSIR 222

April 1, 2022

Problem Statement

Dave's Good Storage Solutions *"We're serious about your data this time"*

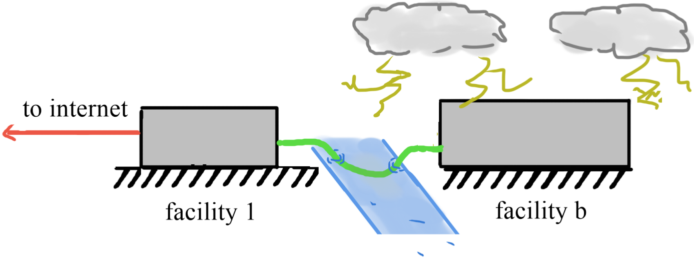
- We store backup data for Reddit at facilities 1 and b
- Facility 1 stores reddit post content for r/haskell, r/lisp
- Facility b stores organizational and identifying data, name of subreddits.



Dave's Good Storage Solutions

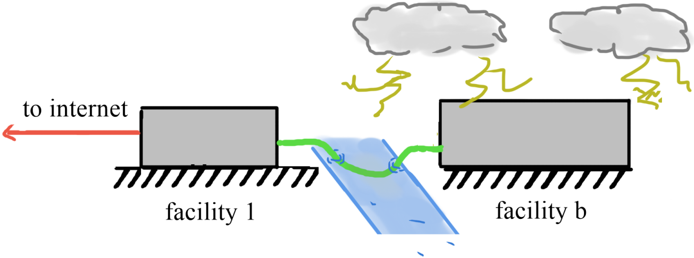
"We're serious about your data this time"

- We store backup data for Reddit at facilities 1 and b
- Facility 1 stores reddit post content for r/haskell, r/lisp
- Facility b stores organizational and identifying data, name of subreddits.



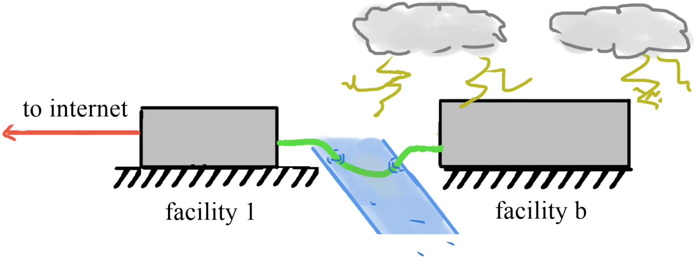
Dave's Good Storage Solutions *"We're serious about your data this time"*

- We store backup data for Reddit at facilities 1 and b
- Facility 1 stores reddit post content for r/haskell, r/lisp
- Facility b stores organizational and identifying data, name of subreddits.



Dave's Good Storage Solutions *"We're serious about your data this time"*

- We store backup data for Reddit at facilities 1 and b
- Facility 1 stores reddit post content for r/haskell, r/lisp
- Facility b stores organizational and identifying data, name of subreddits.



Disaster Planning

Data loss

- What if facility b is lost?
- Reddit will lose valuable data
- DGSS will likely lose Reddit as a customer
- DGSS will lose reputation, and potentially other existing and future customers

Disaster Planning

Data loss

- What if facility b is lost?
- Reddit will lose valuable data
- DGSS will likely lose Reddit as a customer
- DGSS will lose reputation, and potentially other existing and future customers

Disaster Planning

Data loss

- What if facility b is lost?
- Reddit will lose valuable data
- DGSS will likely lose Reddit as a customer
- DGSS will lose reputation, and potentially other existing and future customers

Disaster Planning

Data loss

- What if facility b is lost?
- Reddit will lose valuable data
- DGSS will likely lose Reddit as a customer
- DGSS will lose reputation, and potentially other existing and future customers

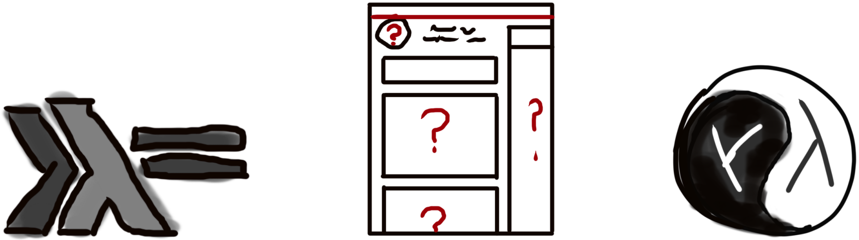
Disaster Planning

Data loss

- What if facility b is lost?
- Reddit will lose valuable data
- DGSS will likely lose Reddit as a customer
- DGSS will lose reputation, and potentially other existing and future customers

Problem statement

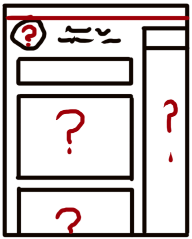
- Given the complete loss of identifying data, can we recover at least the origin subreddits for our post content?
- Given incomplete recovery, how **accurately** can we re-assign those origin subreddits?



r/haskell ⟵ ? unclassified reddit posts ? ⟶ r/lisp

Problem statement

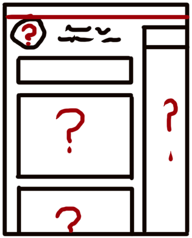
- Given the complete loss of identifying data, can we recover at least the origin subreddits for our post content?
- Given incomplete recovery, how **accurately** can we re-assign those origin subreddits?



r/haskell ⟵ ? unclassified reddit posts ? ⟶ r/lisp

Problem statement

- Given the complete loss of identifying data, can we recover at least the origin subreddits for our post content?
- Given incomplete recovery, how **accurately** can we re-assign those origin subreddits?



r/haskell ⟵ ? unclassified reddit posts ? ⟶ r/lisp

Data

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

Two programming languages

Haskell

https://www.haskell.org/

- First in 1990
- Functional, static typed
- Lazy evaluation

Lisp

https://common-lisp.net/, https://lisp-lang.org/

- Family of languages, first in 1958
- Functional and OOP (class definitions)
- Not lazy

A quick look – defining Quicksort

Haskell looks like...

https://wiki.haskell.org/Introduction

```
qsort (p:xs) = qsort [x | x<-xs, x<p] ++ [p] ++ qsort [x | x<-xs,
x>=p]
```

A quick look – defining Quicksort

Lisp looks like...

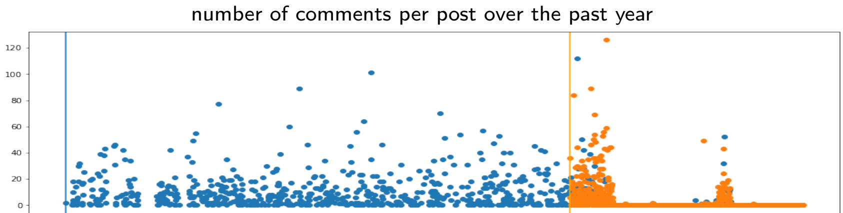
https://wiki.haskell.org/Introduction

```
(defun qs (list)
  (if (cdr list)
    (flet ((pivot (test)
        (remove (car list) list :test-not test)))
    (nconc (qs (pivot #’>)) (pivot #’=) (qs (pivot #’<))))
  list))
```

Data description

Timeframe

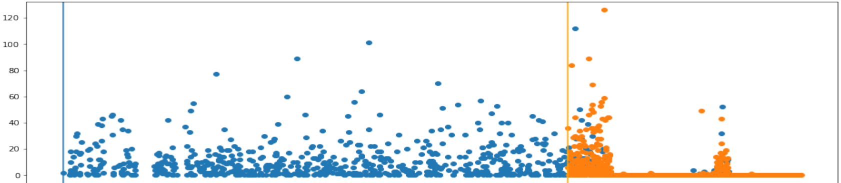
- Data for the latest 1000 submissions to subreddits r/haskell and r/lisp
- r/haskell: November 2021 — March 2022
- r/lisp: February 2021 — March 2022
- Tradeoff between class balance and post age.



Data description

Timeframe

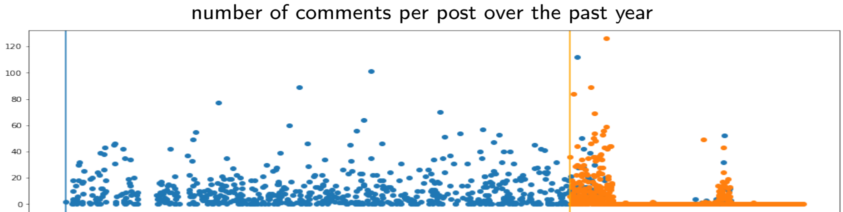
- Data for the latest 1000 submissions to subreddits r/haskell and r/lisp
- **r/haskell:** November 2021 — March 2022
- **r/lisp:** February 2021 — March 2022
- Tradeoff between class balance and post age.


number of comments per post over the past year

Data description

Timeframe

- Data for the latest 1000 submissions to subreddits r/haskell and r/lisp
- **r/haskell:** November 2021 — March 2022
- **r/lisp:** February 2021 — March 2022
- Tradeoff between class balance and post age.

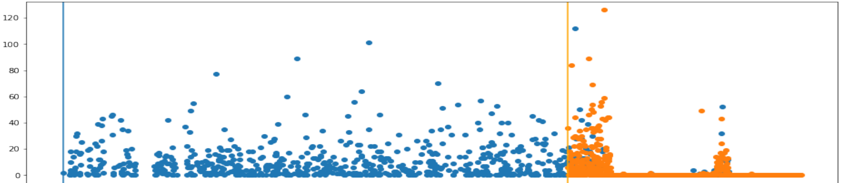


Data description

Timeframe

- Data for the latest 1000 submissions to subreddits r/haskell and r/lisp
- **r/haskell:** November 2021 — March 2022
- **r/lisp:** February 2021 — March 2022
- Tradeoff between class balance and post age.

number of comments per post over the past year



Features

The raw data comes with 82 features, but most were excluded for sparse (*presentation meta: or overly-identifying*) data.

It was reduced to about a quarter of this:

subreddit (target)
domain
is_original_content
is_self
upvote_ratio
author_flair_template_id

selftext (post content)
is_crosspost
is_reddit_media_domain
num_comments
thumbnail_height
poll_data

title
crosspost_subreddit
is_robot_indexable
score
thumbnail_width
post_hint

Feature highlight: links to other sites

is_crosspost

True if the post came with associated crosspost data. A crosspost is a copied post from another subreddit.

crosspost_subreddit

The title of subreddit a crosspost came from.

domain

The domain of the primary link outside of Reddit.

These features were dummified. Feature count: 400+.

Feature highlight: selftext and title

selftext

The primary text of the submission.

title

The title of the submission.

These features were count-vectorized into 1-, 2-, and 3-grams. The number of instances a post had a word (1-gram), sequence of two words (2-gram), or sequence of three words (3-gram), for all 1-3 grams in all of the titles and selftexts.

Feature count: 2700+ All bool or numeric.

Models

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Models

The data was split into training and test sets in a **2:1 ratio.** A number of classification models were fit to the data using `sklearn`.

MultinomialNaiveBayes()

DecisionTreeClassifier()

RandomForest()

SVC() Support Vector Classifier

LogisticRegressionCV() with StandardScaler()

Performance

The best metric for success is accuracy. The baseline accuracy is 0.50.

model	**fit time** (s)	**training accuracy**	**testing accuracy**
MultinomialNaiveBayes()	5.2	0.922	0.923
DecisionTreeClassifier()	2.51	1.0	1.0
SVC()	17.9	0.504	0.505
RandomForest()	6.83	1.0	1.0
LogisticRegressionCV()	36.8	1.0	0.997

Best model so far

DecisionTreeClassifier()
with sklearn default parameters

train/test accuracies: **1.0/1.0**

fit time: **2.5 seconds**

predict time per observation: **752 ms**

The **DecisionTreeClassifier()** splits the data over pivot features, one at a time. There are likely some very powerful features exclusive to each set.

There are also weak words common to both targets.

most common words include:

lisp	common	common lisp	scheme
sbcl	cl	code	programming
emacs	list	question	web
tutorial	clojure	released	python
functions	library	syntax	compiler
game	app		

Next Steps and the Future

Next steps

Comments

- Comments are also stored in our facility
- Comments carry link post id tags, which could be used to reconstruct a comment tree
- Comment trees would reveal a detailed structure of subreddit communication styles

Next steps

Comments

- Comments are also stored in our facility
- Comments carry link post id tags, which could be used to reconstruct a comment tree
- Comment trees would reveal a detailed structure of subreddit communication styles

Next steps

Comments

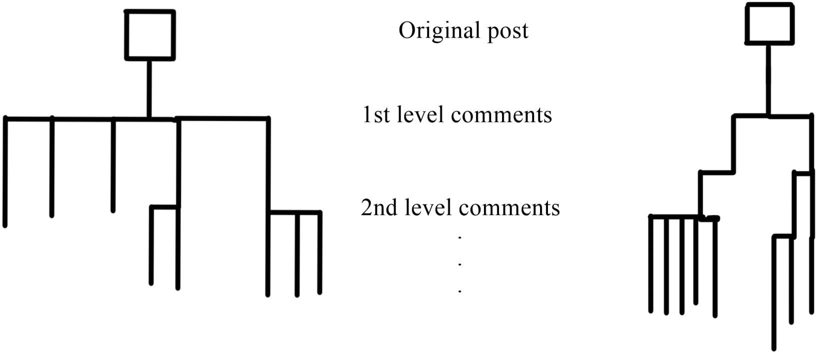
- Comments are also stored in our facility
- Comments carry link post id tags, which could be used to reconstruct a comment tree
- Comment trees would reveal a detailed structure of subreddit communication styles

Next steps

Comments

- Comments are also stored in our facility
- Comments carry link post id tags, which could be used to reconstruct a comment tree
- Comment trees would reveal a detailed structure of subreddit communication styles

Comment trees by hierarchy



Original post

1st level comments

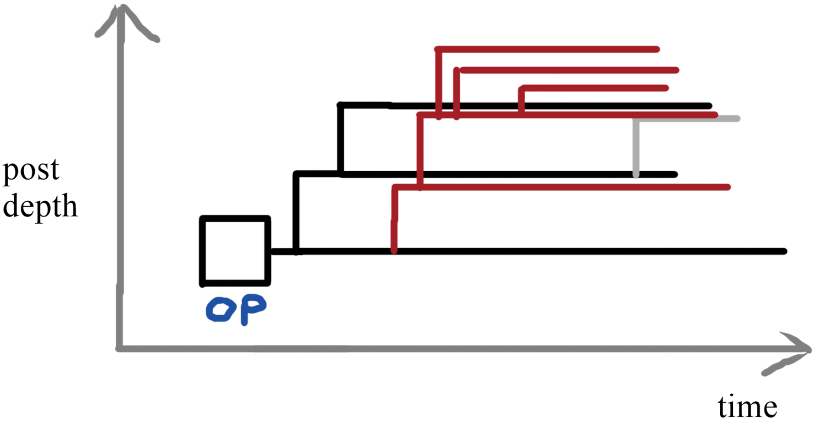
2nd level comments

.

.

.

Comment trees by time



Time

Posting rates and distributions

- Distribution of posts with time carries structural information similar to comment trees

Cultural evolution

- Track domain events, language updates, upsets or achievements, cultural evolution in the community
- Track other communities (Blogs (including Twitter), journals, communities (StackExchange, GitHub) for prominent contributors and keywords
- Re-train the model to reflect these changes from within the subreddit

Time

Posting rates and distributions

- Distribution of posts with time carries structural information similar to comment trees

Cultural evolution

- Track domain events, language updates, upsets or achievements, cultural evolution in the community
- Track other communities (Blogs (including Twitter), journals, communities (StackExchange, GitHub) for prominent contributors and keywords
- Re-train the model to reflect these changes from within the subreddit

Time

Posting rates and distributions

- Distribution of posts with time carries structural information similar to comment trees

Cultural evolution

- Track domain events, language updates, upsets or achievements, cultural evolution in the community
- Track other communities (Blogs (including Twitter), journals, communities (StackExchange, GitHub) for prominent contributors and keywords
- Re-train the model to reflect these changes from within the subreddit

Time

Posting rates and distributions

- Distribution of posts with time carries structural information similar to comment trees

Cultural evolution

- Track domain events, language updates, upsets or achievements, cultural evolution in the community
- Track other communities (Blogs (including Twitter), journals, communities (StackExchange, GitHub) for prominent contributors and keywords
- Re-train the model to reflect these changes from within the subreddit

Time

Posting rates and distributions

- Distribution of posts with time carries structural information similar to comment trees

Cultural evolution

- Track domain events, language updates, upsets or achievements, cultural evolution in the community
- Track other communities (Blogs (including Twitter), journals, communities (StackExchange, GitHub) for prominent contributors and keywords
- Re-train the model to reflect these changes from within the subreddit

Thank you!