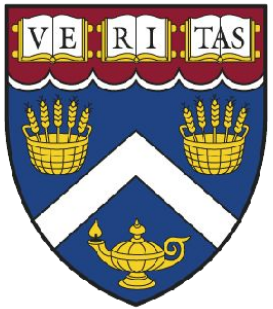
Everything is hypothesis: please correct/guide us if we are wrong!

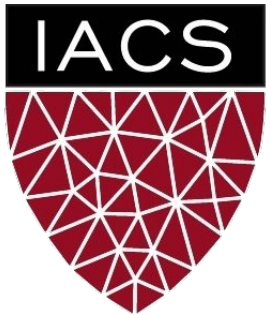
CS109A Final Project

Twitter Bot Detection

Initial Meeting with Adviser



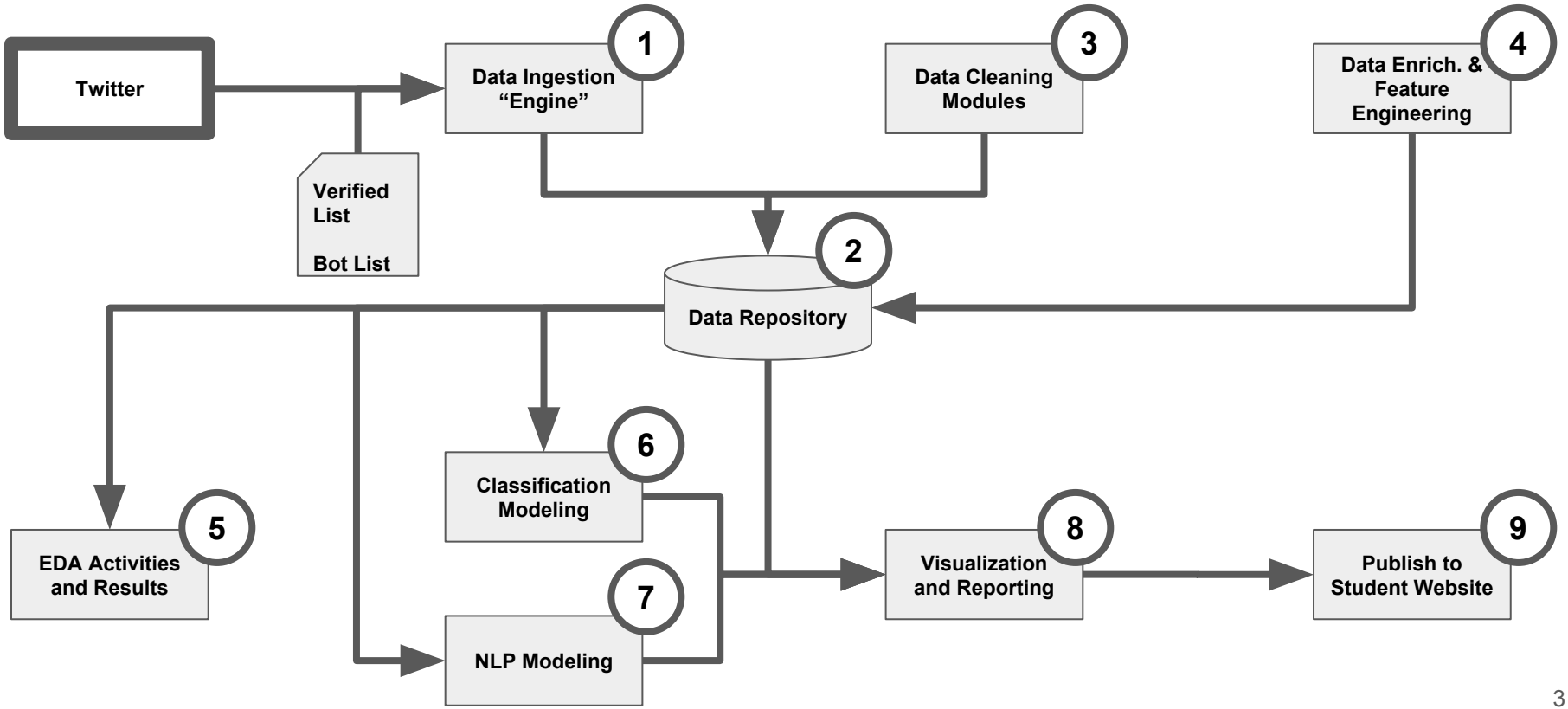
Project Team #6
August 10, 2018



Project Goals

- High Level Design
- Data Schematic: Raw/Source Data
- Data Schematic: Enriched/Engineered Data
- Future Development (i.e., Out-of-Scope)
- Next Steps & 30 Day Project Plan
- Other Questions

High Level Design



Data Schematic: Raw/Source Data

Users:	
Attribute	**Type**
id	Int64
id_str	String
name	String
location	String
url	String
verified	Boolean
followers_count	Int
friends_count	Int
listed_count	Int
favorites_count	Int
statuses_count	Int
created_at	String
utc_offset	null
time_zone	null
language	String

Tweets:	
Attribute	**Type**
user_id	Int64
created_at	String
id	Int64
id_str	String
text	String
source	String
reply_count	Int
retweet_count	Int
favorite_count	Integer
possibly_sensitive	Boolean

Entities	
Attribute	**Type**
tweet_id	Int64
url	URL object

Stretch Goal = Limited Entity Analysis

- We will try to include in unpacking “URL” object and analyzing “boolean” presence and/or “count”
- Deeper diving into URLs and/or unpacking other entities is out of scope
  - Hashtags
  - Media
  - User mentions
  - Symbols
  - Polls

Doable? Acceptable? Alternatives? Suggestions?

- Only 7 day history on standard API
- Twitter limits?
- Collecting all tweets for fixed list of known users: not streaming & not by subject
- Suggestions for Bots? For verified users?
- Missing columns / attributes?
- What is with “pink” attributes? “null”?

Data Schematic: Enriched/Engineered Data

Simple Tweet Analysis

NLP Modeling

Time Bucketing Pre-Process

Tweets:	
Attribute	**Type**
NLP: text quantity	TBD
NLP: text sentiment	TBD
NLP: text complexity	TBD
etc. (based on API?)	TBD
etc. (based on API?)	TBD
Entity count	Int
text_entity_ratio	Float

Does this make sense?

Should we do more or less?

Suggestions?

Time of Day Buckets:	
Attribute	**Type**
user_id	Int64
Minute-of-the-Day	Int
tweet_count	Int
reply_count	Int
retweet_count	Int
favorite_count	Int
possibly_sensitive_count	Int

Or just groupby query on-the-fly?

Future Development (i.e., Out-of-Scope)

- “Networking” from author/tweet to evaluate/weight connected users (e.g., followers, following, friends, etc.)
- “Networking” from author/tweet to evaluate/weight connected entities (e.g., diving into URLs, photos, other embedded entities)
- Even superficial analysis of some/many/all(?) entities (see previous page for stretch goal on entities)

- More than 7 days (not available via standard API
- More than “X” verified accounts and “Y” known bots (any suggestions on X & Y?)
- Non-english tweets

Does this make sense?

Should we do more or less?

Next Steps & 30 Day Project Plan

- **Week of July 15**
  - Create database
  - Load database with initial test bed
  - Create simple (non-NLP) enrichment
  - Create time-bucketing pre-processor
  - Develop more formal EDA suggestions, specifications
  - Start EDA (ad hoc)
  - Identify and play with NLP modeling
  - Finalized bot & not-bot lists
- **Week of July 22**
  - Identify and play with Classification modeling
  - Load full training data
  - Run simple enrichment and time bucketing pre-processor
  - Complete EDA & Milestone #3 (July 27)
- **Week of July 29**
  - Finalize models and analysis
  - Run NLP analysis on training data
  - Load testing data with enrichments
  - Compare test to training results
- **Week of August 5 (ending 8/10/18)**
  - Prepare final presentation on student website

Areas of Focus, though Everyone Has to Help Broadly

Eumar	Data ingestion, repository and technical tools/architecture
Andrew	EDA and Classification modeling
Jason	EDA and NLP modeling
Mark	Scrum master, data cleaning/enrichment, final reporting, student website

Other Questions

- How do we do all of that while doing homeworks, working, etc.

- Re-using existing NLP tools? Suggestions?
- Building and/or re-using existing Classification tools? Suggestions?
- Skype, GitHub...other collaboration tools?
- MySQL and Azure...other technical architecture suggestions?
- Does this do what we want?
  - **`API.user_timeline`**(**[***id/user_id/screen_name*****][****, since_id*****][****, max_id*****][****, count*****][****, page*****]**)
  - Returns the 20 most recent statuses posted from the authenticating user or the user specified. It's also possible to request another user's timeline via the id parameter.