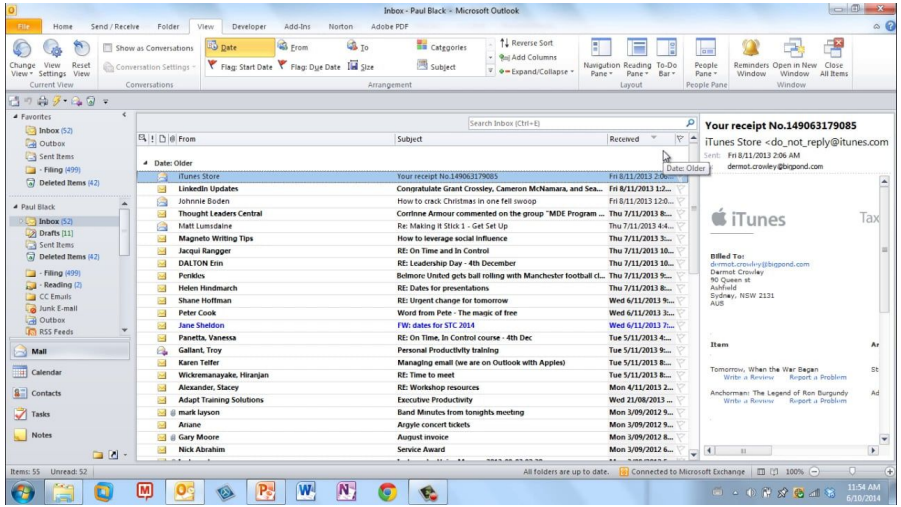
# Smart Folders for Work Email

Danny Strockis
Adarsh Ramakrishnan
Grace Lin
Rutika Banakar
March 2018

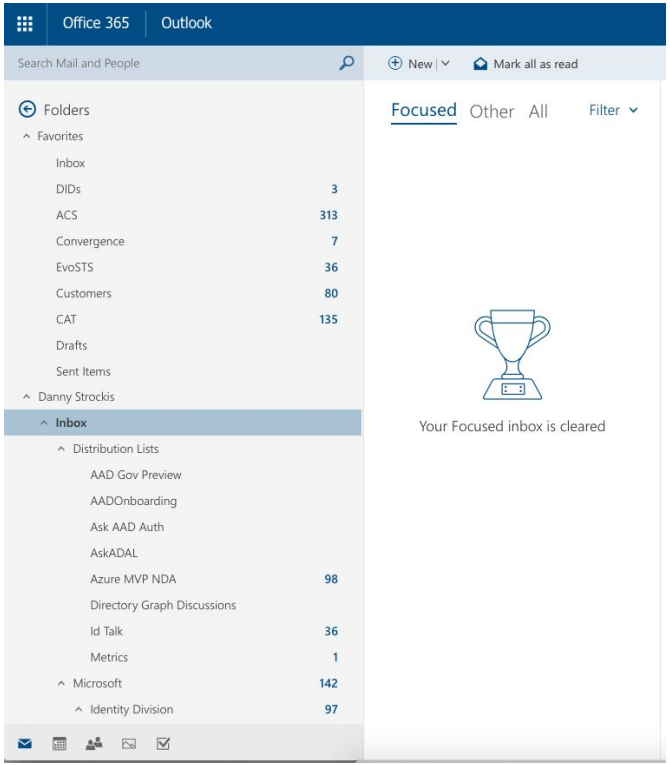
# The problem with work email

- Modern office employees receive hundreds of emails per day (average ~120)
- Processing emails accounts for one third of time spent working
- Email aggregates many information sources - messages from coworkers, articles, routine status, customer inquiries, etc.
- Email processing techniques:
    - Leave everything in inbox - process sequentially by time received
    - Sort by topics - sort emails into folders as you read
    - Inbox rules - sort/delete emails via fixed logic




- The problem: information overload creates an endless time sink, and lack of organization makes it impractical to identify the most important emails and ignore the rest.

# A better approach




- Automatically sort email in the way that works for you:
  - By project
  - By sender
  - By topic
  - By type
  - …
- Focus on the folders that of higher importance, check others infrequently
- Our project:
  - Find ways of automatically organizing emails to help people save time

Approach #1: auto-sort emails into user defined folders based on previous sorting patterns
Approach #2: suggest folders to use automatically

# Expected challenges

- Data availability:
  - Companies have email privacy requirements with varying strictness
  - Companies have varying email retention policies
  - Companies have various security policies (namely, encryption at rest)
- Data formats:
  - Emails come from various providers (Office 365, Gmail, On-prem email servers)
  - Not all providers leverage the same schema (ex: conversations)
- Data ingest:
  - Sync’ing data out of servers is non trivial
  - Not all emails are interesting
- Variety in user behavior:
  - Everyone sorts their inboxes in their own custom ways, it will be difficult to make a one-size fits all model
- If we use per-user models, we will need a lot of data per user (average ~30K emails/year)

# Data Sources & Cleaning

- Data Sources Used
  - The Enron email dataset contains approximately 517,401 emails generated by employees of the Enron Corporation. It was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse.
  - The data set contains 2 columns:
    - File: this column contains sender information
    - Message: this column contains various features

```
Message-ID: <15464986.1075855378456.JavaMail.evans@thyme>
Date: Fri, 4 May 2001 13:51:00 -0700 (PDT)
From: phillip.allen@enron.com
To: john.lavorato@enron.com
Subject: Re:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: John J Lavorato <John J Lavorato/ENRON@enronXgate@ENRON>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\'Sent Mail
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst

Traveling to have a business meeting takes the fun out of the trip.  Especially if you have to prepare a p
resentation.  I would suggest holding the business plan meetings here then take a trip without any formal
business meetings.  I would even try and get some honest opinions on whether a trip is even desired or nec
essary.
```

# Data Sources & Cleaning

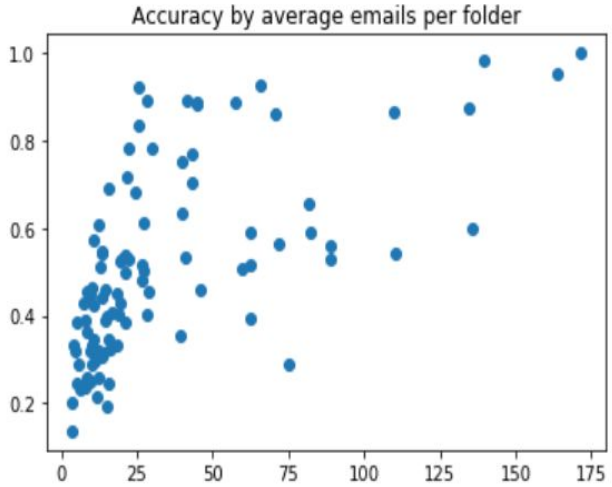
- Data Cleansing
  - Extract different features from messages: sender, recipient, cc, bcc, subject, origin, date, message id, folders, and body
  - Clean and tokenize data, including removing stop words, numbers, and structural words
  - Filter computer-generated folders, including sent_mail, all_documents, deleted_items, inbox, discussion_threads, notes_inbox, sent_items and sent
  - Combine sender, recipient, cc, and bcc as users
  - Filter to exclude users with fewer than 100 emails
  - Results:
    - Reduce dataset from 517K to 73K
    - Reduce users from 267 to 97

# Feature Engineering & Supervised Learning

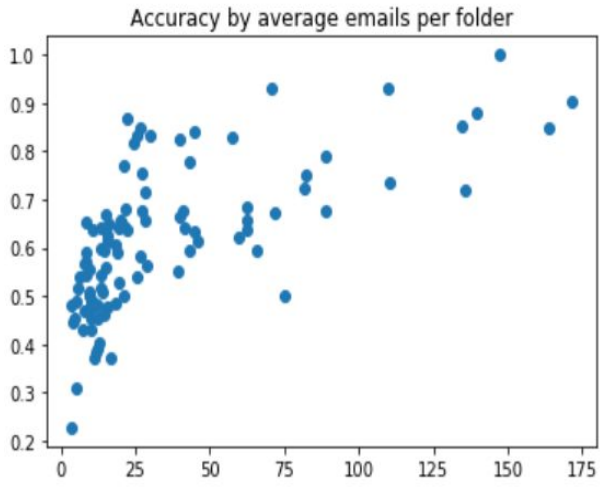
- Feature Engineering
  - Features we use: senders, recipients, subjects, and email bodies
  - Transformations:
    - Perform bag of words vectorization on Subject
    - Convert Subject from bools to tf-idf values
    - Convert people column into email address counts
    - Merge people counts with subject tf-idf

- Supervised Learning
  - Per-user logistic regression
    - Average accuracy for Logistic Regression: 0.51
  - Per-user naive bayes
    - Average accuracy for Naive Bayes: 0.61

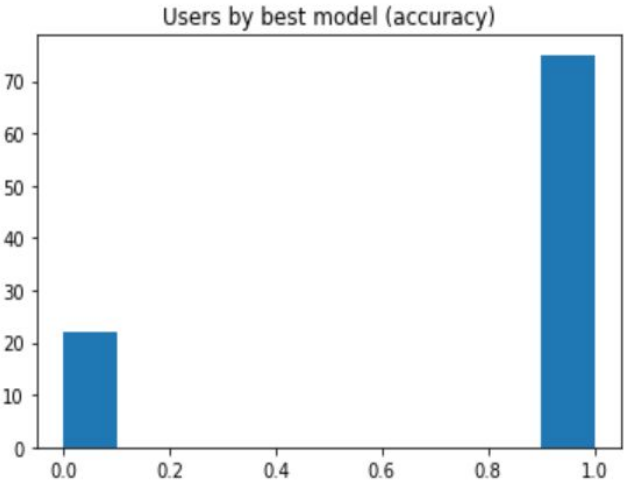
# Feature Engineering & Supervised Learning




Logistic Regression




Naive Bayes




LR V.S. NB

# Unsupervised Learning

## Learning Model

- Used the same data cleaning methods as in supervised learning.
- Used the Bag-of-Words representation of email bodies (or documents) to train a Latent Dirichlet Allocation (LDA) model, a topic model, to learn 10 topics from all the documents.
- Then for each given document BoW, topic distribution was obtained and the topic with highest probability was assigned as the topic for that particular document.
- Used the topic labels to come up with the top three most frequently occurring topics in the email body corpus

# Conclusion

- The Supervised Learning method worked well for predicting the folder for emails on a per-user basis. Whereas the unsupervised model worked well for exploring the dataset and discovering potential folder names based on topics.
- LDA topic model helped us analyze the various topics among all the emails.
- But using a combination of Logistic Regression and Naive Bayes to predict a user's email sorting behavior gave us the best results.

Next steps:

- Use the topics given by LDA as labels for the emails/documents and train a model to predict folders for incoming emails.
- Random Forest classifier might work best for this use case.

Thank you!

Questions?