Constructing the pipeline

Production pipeline outline

Informed by our experiments, here is the outline of our final production pipeline:

1. The pipeline first deduplicates the emails by hash. This is estimated to cut down the number of emails from 50k to 25k.
2. The pipeline then runs CPU-based Presidio to identify emails without PII. This has an FNR of 0% and thus extremely high confidence. This will cut down number of emails from 25k to 13k.
3. The pipeline then runs a Presidio-augmented 5-shot Haiku classifier. This classifier takes in the email, the entities flagged by Presidio, and 5 examples from a human labelled dataset, and outputs what PII categories are present, if any.
4. The pipeline then collects the results and outputs the following deliverables:
    a. A table (.csv) that can be easily filtered and contains the following columns
        i. `ID`, integer, this is the original email ID
        ii. `duplicated`, boolean, this is whether the email is duplicated. For a set of duplicated emails, exactly one will be processed in the remaining of the pipeline.
        iii. `pii_absent_high_confidence`, boolean, this is set to 1 if Presidio determines that there is no PII in the email. Presidio has FNR=0, and we use a confidence threshold of 0.
        iv. `pii_absent_low_confidence` , boolean, this is set to 1 if Haiku does not flag any of the PII variants as present.
        v. We then

    At the end, we want to create a CSV that has the cols
    ID, duplicated, pii_absent_high_confidence, pii_absent_low_confidence, each of the PII variant gets a boolean column

duplicated is from the deduplication
pii_absent_high_confidence is boolean from presidio
pii_absent_low_confidence is boolean from the haiku
and each of the pii variant is also from haiku

The final output has

Sample run

We ran the pipeline on the 9k sample.

Metrics

1. **Deduplication**
    - Input: 9,199 emails
    - Reduced 9,199 → 4,712 unique emails (48.8% reduction)
    - Processed in 0.71s
2. **Presidio detection**
    - Input: 4,712 emails (51.2% of original 9k sample)
    - 3,672 emails with PII (77.9%)
    - 1,040 emails without PII (22.1%)
    - Time: 5.6 minutes (337.97s) for 4,712 emails (we can make this <1 minute on an EC2)
    - Cost: $0 (local on my macbook, <$1 on EC2)
3. **Haiku classification**
    - Input: 3,672 emails (39.9% of original 9k sample)
    - 2,976 emails with PII (63.2%)
    - 1,736 emails without PII (36.8%)

    - Time: 11.2 minutes (we can make this <5 minutes by increasing requests limits, we currently use a limit of 10)
    - Cost: $19.88 total ($0.0042 per email)

4. **Final output**

- CSV with 9,199 rows (all original emails, duplicates inherit results)
- 19 columns (ID, flags, + 15 PII type booleans)

Client next-steps

I would recommend the client to do a manual review of the 3,672 emails flagged by Presidio (filter by `pii_absent_high_confidence`). Within this group, we flagged 63% of them that likely contains PII, and 37% that likely doesn't (filter by `pii_absent_low_confidence`). I will also tell the client that the low confidence classification has a 60% exact match accuracy to the PII categories specified, and that the client should pay attention to address, driver's license and full name when commissioning manual review since this are less likely to be flagged by our pipeline.

Time breakdown

- Total time: ~17 minutes (1,009.66s)
- Presidio: 33.5% of time
- Haiku: 66.3% of time
- Other stages: <1%
- Note: we can make this <5 minutes by running on EC2 (Presidio 6m → 1m) and tripling Haiku limits from 10 to 30 (11m → 3m)

Further work

We can further optimize for time. Suppose we have a 50k dataset, and deduplication gives us 25k emails in <1 min.

- [ ] Presidio currently takes 6 minutes to process 5k emails on a MacBook Air, giving 72s/1k email. We can spin up an EC2 machine with stronger CPU to host the pipeline and cut this down to 10 seconds/1k emails. This will give 5 minutes for the 25k emails.

- [ ] Now suppose after Presidio we cut down emails by 1/4, giving us 20k emails. We currently parallelize API requests to Haiku at 10 concurrent requests, resulting in also around 10 emails/second. We can reasonably scale this to 40 emails/second or better. This will give 9 minutes for the 20k emails.

Once implemented, this will give us <15 minutes total for 50k emails, with a cost of around $110.

Appendix

```
PII TYPE BREAKDOWN (Haiku):
  email address                  2,078 ( 44.1% of emails)
  full name                      1,678 ( 35.6% of emails)
  phone number                   1,455 ( 30.9% of emails)
  password                          61 (  1.3% of emails)
  username                          48 (  1.0% of emails)
  ssn                                4 (  0.1% of emails)
  bank account #                     4 (  0.1% of emails)
  last 4                             3 (  0.1% of emails)
  credit card #                      3 (  0.1% of emails)
  student identification #           2 (  0.0% of emails)
  tin                                2 (  0.0% of emails)
```