

# [PRD] SSN Sampler

**Summary:** Samples SSN occurrences in unstructured data.

|  |  |
| --- | --- |
| **Created:** Jun 1, 2021  **Owner:** mail@ashwin.live  **Contributors:** | **Status:** WIP | **In Review** | Approved | Obsolete  **RFC:** Link to RFC when created |



The SSN Scanner samples a location (s3 bucket) for SSN occurrences and reports it to the analyst to see the extent of exposure. The analyst will be able to configure the accuracy required in the system output.

## Background

We are given a storage location (say S3), and we need to find the number of documents in the location having SSN numbers. This allows us to have an idea of the exposure liability of storing PII data in cloud storage models.

## Problem

1. Cloud storage locations having N number of text documents (txt/csv/pdf)
2. The documents could have exposed SSN numbers in them, thus exposing PII data
3. We need to give an analysis of the exposure by sampling the documents and testing for SSN numbers
4. Trade-off parameters between accuracy and speed (sampling all documents will take too long)
5. Output to denote sampling rate and confidence in findings

## Requirements and Phases

|  |  |
| --- | --- |
|  | **Requirements** |
| **Phase 1:** Setup SSN classification model. Ability to handle pdf/docx formats | SSN classifier should be able to detect SSNs and provide a degree of confidence about the results  Text extraction capabilities to handle pdf/docx formats |
|
| **Phase 2:** Build Sampler Algorithm to account for domain data | Sampling algorithm to handle possible domains (e.g. health data, resumes, financial transactions, chat data) and have elegant switch over on sampling rates |
| **Phase 3:** Integration | Integrate components and build an end-to-end tool for SSN Identification and Analysis |

**Setup SSN classification model**

#### Ability to Detect SSNs

This phase will involve building a ML model that is able to detect SSNs in any document with a high degree of confidence. The model can be off-the-shelf (transfer learning) or a custom model running on a container and is callable via an API.

* Pipeline to convert pdf/docx to text format
* Model should take an input of text and return the SSN IDs present in the document.
* Model should return a confidence level associated with the detected SSN.
* The threshold for detection of the SSN should be user configurable.

##### Acceptance Criteria.

1. Extract text from pdf/docx formats
2. Detect SSNs in text input
3. Confidence score for each SSN

##### Considerations

1. Do we need to handle files other than docx/pdf/txt ?
2. How do you handle the presence of multiple SSN in a file?
3. Do we need to handle the use case of image files like scanned documents?

**Build Sampler algorithm**

#### Sampling Algorithm

This phase will involve building a sampling algorithm that will report out the analysis of the documents along with a confidence measure on the total number of documents. The algorithm should balance between accuracy and speed, given that there might be millions of documents to test.

* Algorithm should take an input of the s3/location folder, along with an optional confidence parameter.
* Algorithm will decide on the sampling rate on the basis of confidence parameters or default to values pre-assigned.
* Algorithm will report out the SSNs detected along with a confidence measure on the result.

##### Acceptance Criteria.

1. Algorithm should not have to sample 100% of documents to arrive at a result
2. Algorithm should accurately give a confidence measure (along with explanation if needed)

##### Considerations

1. Do we need to limit based on the number of documents to be scanned? E.g. 10% of 1MN documents is 100K calls to the SSN classifier model
2. Do we need to keep the sampling rate user configurable?

## User Research

We can carry out user research to understand the different domains, and documents related to the domain. This will give us a better understanding of the sampling rates to be used on a per domain basis. E.g. financial transaction records might have an higher occurrence of SSN in documents relative to Chat transactions.

## Approvals

Red names still require approval, green names with a checkmark have approved.

Each of the representative stakeholders must approve the PRD before it proceeds to kick-off and RFC writing. Sign-off by stakeholders requires these to be satisfied:

* The release summary defines which acceptance criteria are in scope.
* Engineering and Product Management agree on the target release.
* Provides sufficient clarity to author the RFC.
* Schedule a meeting with engineering and design to review the PRD.

**Approved names will have a ✅**

Project Engineering Lead: XX

Product Manager: **✅** XX

VP of Product: XX

Sales Engineer Lead: XX

ProductDesign Lead: XXX

[Add other approvers as necessary]