

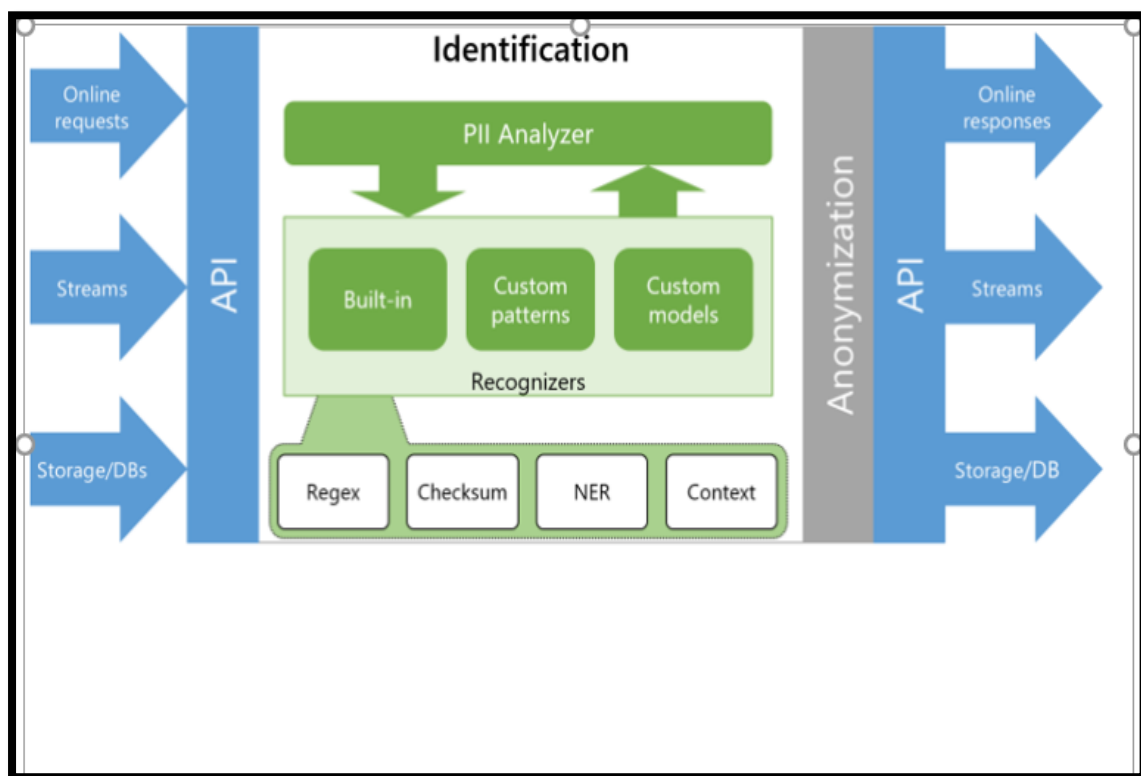
## Problem Statement:

Work on anonymizing using presidio. For example, if there is an email, then it should automatically detect PII information and encrypt/mark it.

## Algorithms:

**PII** stands for Personally Identifiable Information and for detecting this we have Presidio which allows any user to create standard and transparent processes for anonymizing PII entities on structured and unstructured data.

To do so, it exposes a set of predefined PII recognizers (for common entities like emails, credit card numbers and phone numbers), and tools for extending it with new logic for identifying more specific PII entities.



Here we see that after loading the request or files from the DBs, it is sent to PII analyzer, which has several built-in models and custom

patterns and we can also make custom models using machine learning as well.

After analyzer using recognizer through various ways like regex, checksum, it sends its output to an anonymizer which tells or encrypts or marks the PII and sends it back to the API,s

Today I dealt with understanding analyzer part using the Recognizer and I used the pattern recognizer and in that I used regex to identify the email ids provided in that.

## **Code for Analyzer:**

### **Import statements for analyzers:**

- pip install presidio\_analyzer
- pip install presidio\_anonymizer
- from typing import List
- import pprint
- from presidio\_analyzer import AnalyzerEngine, PatternRecognizer, EntityRecognizer, Pattern, RecognizerResult
- from presidio\_analyzer.recognizer\_registry import RecognizerRegistry
- from presidio\_analyzer.nlp\_engine import NlpEngine, SpacyNlpEngine, NlpArtifacts

### **1. Define the regex pattern in a Presidio `Pattern` object:**

- Email\_pattern = Pattern(name="Email\_pattern", regex="[a-zA-Z0-9+.\_-]+@[a-zA-Z0-9.\_-]+\.[a-zA-Z0-9\_-]+", score = 0.5)

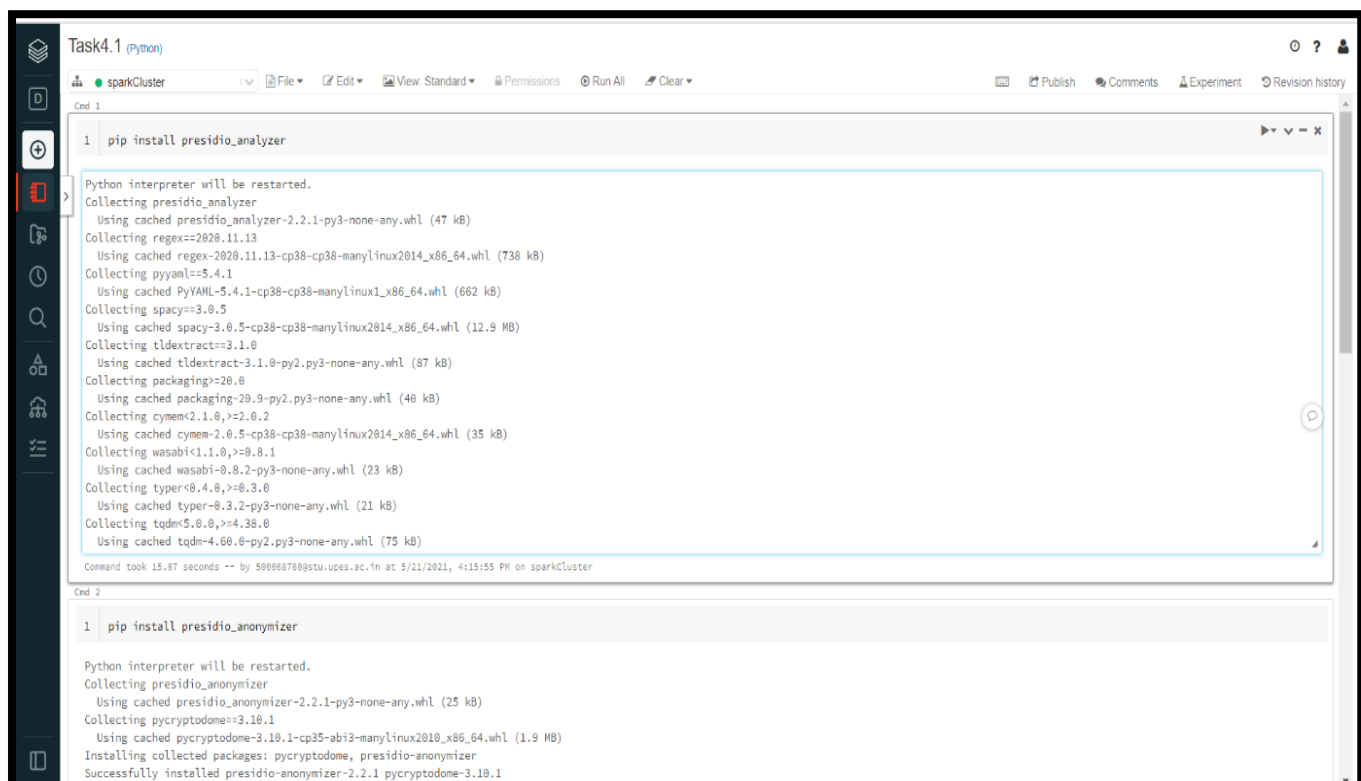
### **2. Define the recognizer with one or more patterns**

- Email\_recognizer =  
PatternRecognizer(supported\_entity="Email", patterns =  
[Email\_pattern])

### 3. Testing the analyzer:

- myemail = "Feel free to mail me the issue at  
lakshay.sharma@rani.ai or to the our head  
aparnesh.gaurav@rani.ai"
- Email\_result = Email\_recognizer.analyze(text=myemail,  
entities=["Email"])
- print("Result:")
- print(Email\_result)

### Snapshots:



The screenshot shows a Jupyter Notebook interface with two terminal outputs. The first terminal, labeled 'Task4.1 (Python)', shows the command 'pip install presidio\_analyzer' and its output, which lists various dependencies and their versions. The second terminal, labeled 'Task4.2', shows the command 'pip install presidio\_anonymizer' and its output, which lists dependencies and their versions.

```
Task4.1 (Python)
1 pip install presidio_analyzer

Python interpreter will be restarted.
Collecting presidio_analyzer
  Using cached presidio_analyzer-2.2.1-py3-none-any.whl (47 kB)
Collecting regex==2020.11.13
  Using cached regex-2020.11.13-cp38-cp38-manylinux2014_x86_64.whl (738 kB)
Collecting pyyaml==5.4.1
  Using cached PyYAML-5.4.1-cp38-cp38-manylinux1_x86_64.whl (662 kB)
Collecting spacy==3.0.5
  Using cached spacy-3.0.5-cp38-cp38-manylinux2014_x86_64.whl (12.9 MB)
Collecting tldextract==3.1.0
  Using cached tldextract-3.1.0-py2.py3-none-any.whl (87 kB)
Collecting packaging>=20.0
  Using cached packaging-20.9-py2.py3-none-any.whl (40 kB)
Collecting cymem<2.1.0,>=2.0.2
  Using cached cymem-2.0.5-cp38-cp38-manylinux2014_x86_64.whl (35 kB)
Collecting wasabi<1.1.0,>=0.8.1
  Using cached wasabi-0.8.2-py3-none-any.whl (23 kB)
Collecting typer<0.4.0,>=0.3.0
  Using cached typer-0.3.2-py3-none-any.whl (21 kB)
Collecting tqdm<5.0.0,>=4.38.0
  Using cached tqdm-4.60.0-py2.py3-none-any.whl (75 kB)

Command took 15.87 seconds -- by 500660769@stu.upes.ac.in at 5/21/2021, 4:15:55 PM on sparkCluster

Task4.2
1 pip install presidio_anonymizer

Python interpreter will be restarted.
Collecting presidio_anonymizer
  Using cached presidio_anonymizer-2.2.1-py3-none-any.whl (25 kB)
Collecting pycryptodome==3.10.1
  Using cached pycryptodome-3.10.1-cp35-abi3-manylinux2010_x86_64.whl (1.9 MB)
Installing collected packages: pycryptodome, presidio-anonymizer
Successfully installed presidio-anonymizer-2.2.1 pycryptodome-3.10.1
```

Task4.1 (Python)

sparkCluster

Collecting tqdm<5.0.0,>=4.38.0  
Using cached tqdm-4.60.0-py2.py3-none-any.whl (75 kB)

Command took 15.87 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:15:55 PM on sparkCluster

Cmd 2

```
1 pip install presidio_anonymizer
```

Python interpreter will be restarted.  
Collecting presidio\_anonymizer  
Using cached presidio\_anonymizer-2.2.1-py3-none-any.whl (25 kB)  
Collecting pycryptodome==3.10.1  
Using cached pycryptodome-3.10.1-cp35-abi3-manylinux2010\_x86\_64.whl (1.9 MB)  
Installing collected packages: pycryptodome, presidio-anonymizer  
Successfully installed presidio-anonymizer-2.2.1 pycryptodome-3.10.1  
WARNING: You are using pip version 20.2.4; however, version 21.1.1 is available.  
You should consider upgrading via the '/local\_disk0/.ephemeral\_nfs/envs/pythonEnv-35de409a-5c76-47f5-b6df-4b21a59e2b63/bin/python -m pip install --upgrade pip' command.  
Python interpreter will be restarted.

Command took 5.51 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:16:48 PM on sparkCluster

Cmd 3

```
1 from typing import List
2 import pprint
3
4 from presidio_analyzer import AnalyzerEngine, PatternRecognizer, EntityRecognizer, Pattern, RecognizerResult
5 from presidio_analyzer.recognizer_registry import RecognizerRegistry
6 from presidio_analyzer.nlp_engine import NlpEngine, SpacyNlpEngine, NlpArtifacts
```

Command took 0.68 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:18:27 PM on sparkCluster

Task4.1 (Python)

sparkCluster

Command took 0.68 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:18:27 PM on sparkCluster

Cmd 4

```
1 # Define the regex pattern in a Presidio 'Pattern' object:
2 Email_pattern = Pattern(name="Email_pattern", regex="[a-zA-Z0-9+_-]*@[a-zA-Z0-9+_-]*\.[a-zA-Z0-9+_-]*", score = 0.5)
3
4 # Define the recognizer with one or more patterns
5 Email_recognizer = PatternRecognizer(supported_entity="Email", patterns = [Email_pattern])
```

Command took 0.83 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:24:05 PM on sparkCluster

Cmd 5

```
1 myemail = "Feel free to mail me the issue at lakshay.sharma@rani.ai or to the our head aparnesh.gaurav@rani.ai"
2
3 Email_result = Email_recognizer.analyze(text=myemail, entities=["Email"])
4 print("Result:")
5 print(Email_result)
```

Result:  
[type: Email, start: 34, end: 56, score: 0.5, type: Email, start: 76, end: 99, score: 0.5]

Command took 0.04 seconds -- by 500066760@stu.upes.ac.in at 5/21/2021, 4:33:26 PM on sparkCluster

Cmd 6

```
1
```

Will next work on anonymization of it.