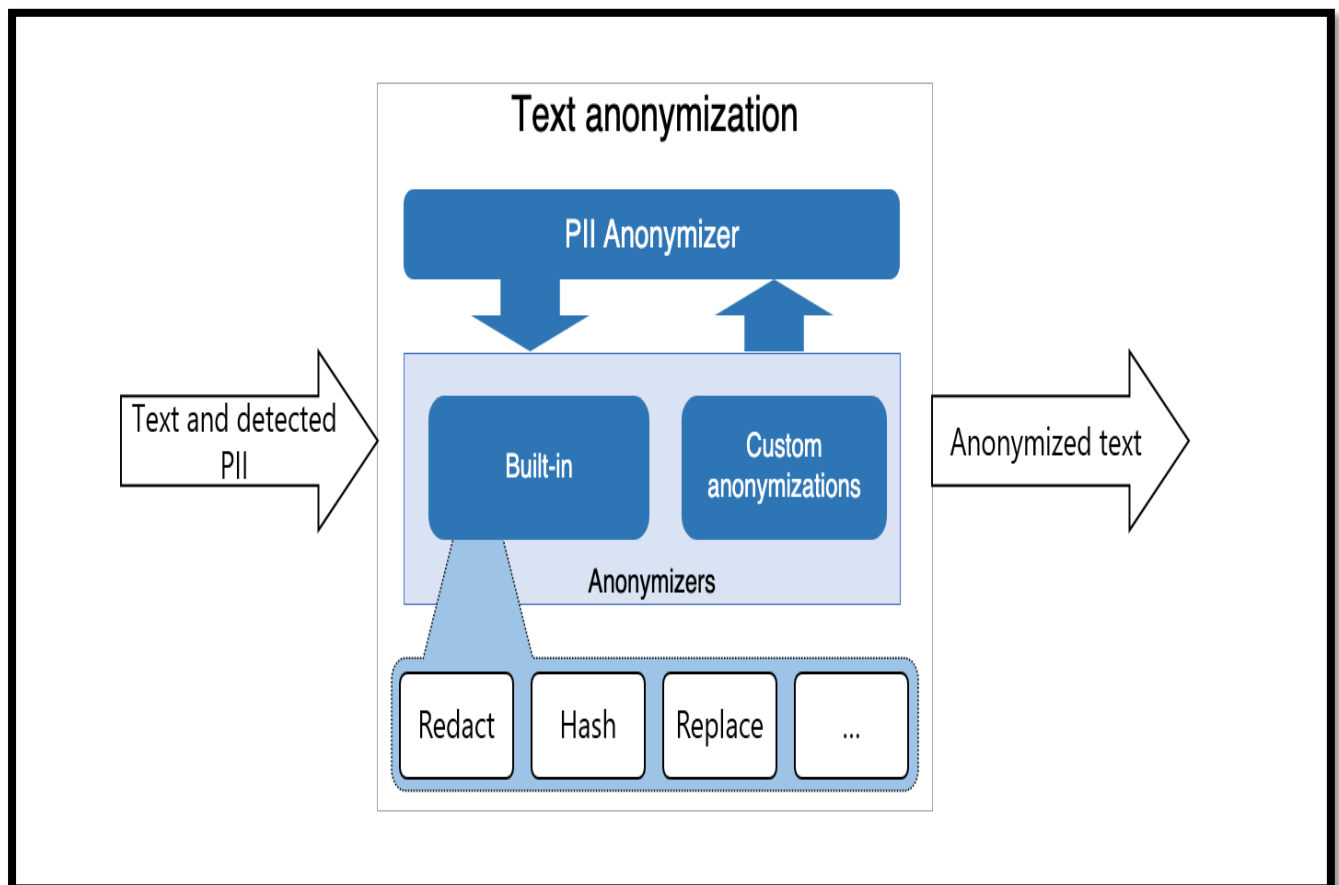


Problem statement:

Work on anonymizing using presidio. For example, hello I am taking disprin and my age is 25 and i am having bipolar disorder then it should automatically detect PII information and encrypt/mark it.

Algorithms:



Completed the Analysing and performing with recognizers and now working on Anonymization, here it takes the text and the already detected PII in the analyzation phase and uses the built anonymizations such as redact, hash, replace and several others.

while it also offers us to create our own custom anonymizations like deploying the machine learning models for that. And that as a result it gives as the anonymized text.

Code:

Import statements for anonymizers:

- from presidio_anonymizer import AnonymizerEngine
- from presidio_anonymizer.entities import RecognizerResult
- from presidio_anonymizer.entities.engine import AnonymizerResult, OperatorConfig

1. Initialising the engine

- engine = AnonymizerEngine()

2. defining the function to anonymize the PII

```
def anonymize_pii(text: str)-> str:
```

```
    # Anonymizers config to define the anonymization type.
```

```
    anonymized_data = engine.anonymize(
```

```
        text=text,
```

```
        analyzer_results=[RecognizerResult("MEDICINE", 18, 26, 0.8),
```

```
                           RecognizerResult("AGE", 41, 43, 0.8),
```

```
                           RecognizerResult("DISEASE", 60, 76, 0.8)],
```

```
        operators={"MEDICINE": OperatorConfig("replace",  
{"new_value": "MEDICINE"}),
```

```
                   "AGE": OperatorConfig("replace", {"new_value":  
"AGE"}),
```

```
                   "DISEASE": OperatorConfig("replace", {"new_value":  
"DISEASE"})}
```

```
    )
```

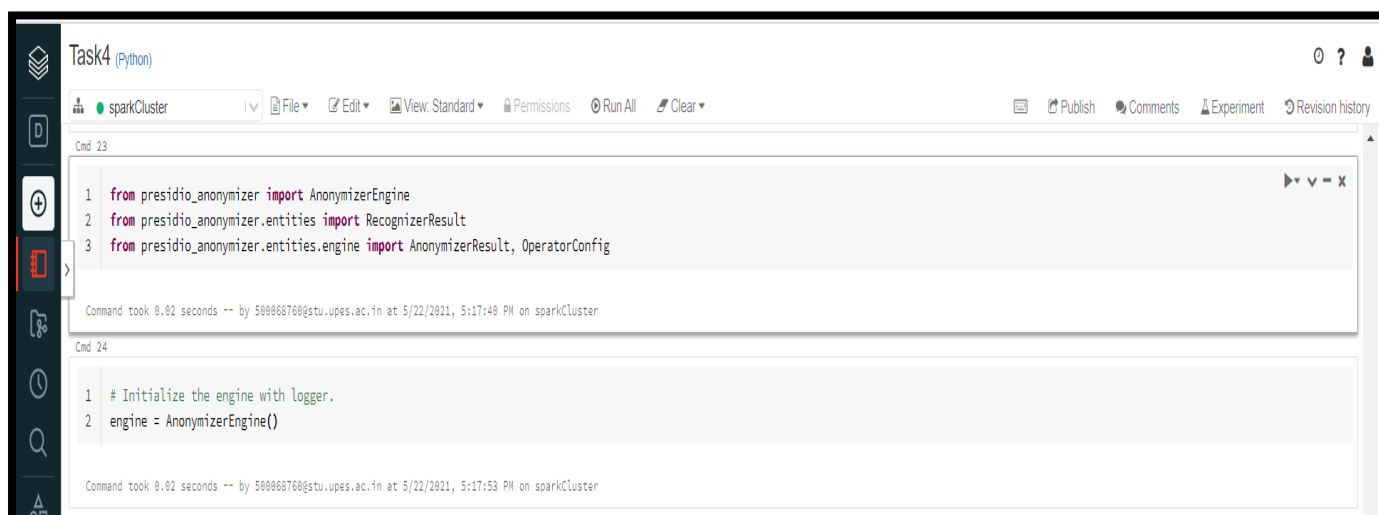
```
    return anonymized_data
```

3. Calling the anonymization by passing our required text “hello I am taking dispring and my age is 25 and i am having bipolar disorder”

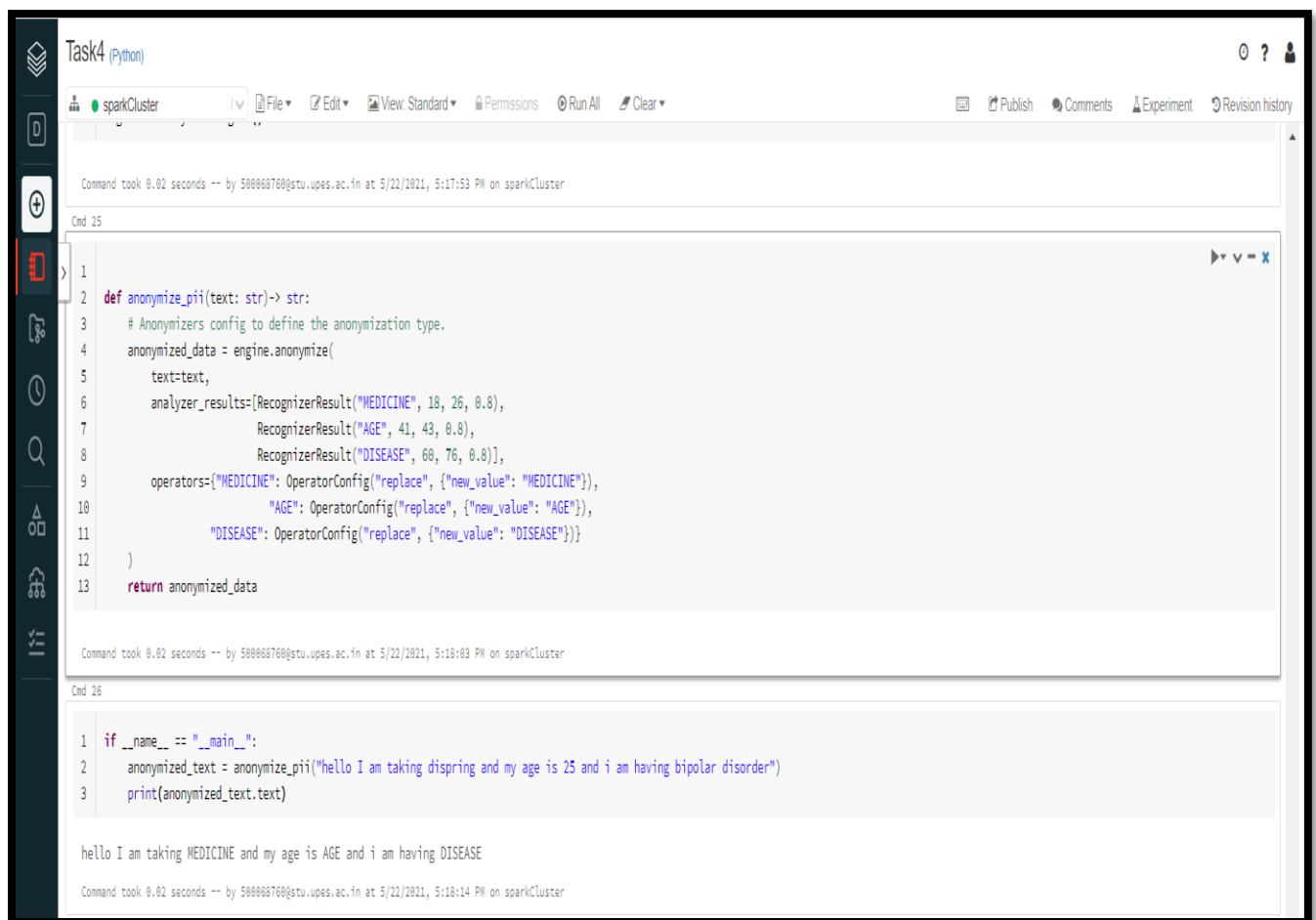
```
if __name__ == "__main__":  
    anonymized_text = anonymize_pii("hello I am taking dispring and  
    my age is 25 and i am having bipolar disorder")  
    print(anonymized_text.text)
```

I have been reading on Name Entity Recognition in presidio but that appears to be available for some predefined types only such that Dates, location, perso, credit card number so then I identified to use python libraries to extract the desired keywords and put into the analyzers. So for this I have been learning on spacy and various other and till be working on that.

Snapshots:



```
Task4 (Python)  
sparkCluster | File | Edit | View: Standard | Permissions | Run All | Clear  
Cmd 23  
1 from presidio_anonymizer import AnonymizerEngine  
2 from presidio_anonymizer.entities import RecognizerResult  
3 from presidio_anonymizer.entities.engine import AnonymizerResult, OperatorConfig  
Command took 0.02 seconds -- by 500066760@stu.upes.ac.in at 5/22/2021, 5:17:40 PM on sparkCluster  
Cmd 24  
1 # Initialize the engine with logger.  
2 engine = AnonymizerEngine()  
Command took 0.02 seconds -- by 500066760@stu.upes.ac.in at 5/22/2021, 5:17:53 PM on sparkCluster
```



```
Task4 (Python)
sparkCluster
File Edit View: Standard Permissions Run All Clear
Publish Comments Experiment Revision history

Command took 0.02 seconds -- by 500060760@stu.upes.ac.in at 5/22/2021, 5:17:53 PM on sparkCluster

Cmd 25
1
2 def anonymize_pii(text: str)-> str:
3     # Anonymizers config to define the anonymization type.
4     anonymized_data = engine.anonymize(
5         text=text,
6         analyzer_results=[RecognizerResult("MEDICINE", 18, 26, 0.8),
7                             RecognizerResult("AGE", 41, 43, 0.8),
8                             RecognizerResult("DISEASE", 60, 76, 0.8)],
9         operators={"MEDICINE": OperatorConfig("replace", {"new_value": "MEDICINE"}),
10                  "AGE": OperatorConfig("replace", {"new_value": "AGE"}),
11                  "DISEASE": OperatorConfig("replace", {"new_value": "DISEASE"})}
12     )
13     return anonymized_data

Command took 0.02 seconds -- by 500060760@stu.upes.ac.in at 5/22/2021, 5:18:03 PM on sparkCluster

Cmd 26
1 if __name__ == "__main__":
2     anonymized_text = anonymize_pii("hello I am taking dispring and my age is 25 and i am having bipolar disorder")
3     print(anonymized_text.text)

hello I am taking MEDICINE and my age is AGE and i am having DISEASE

Command took 0.02 seconds -- by 500060760@stu.upes.ac.in at 5/22/2021, 5:18:14 PM on sparkCluster
```

How to related NLP with AWS Lex

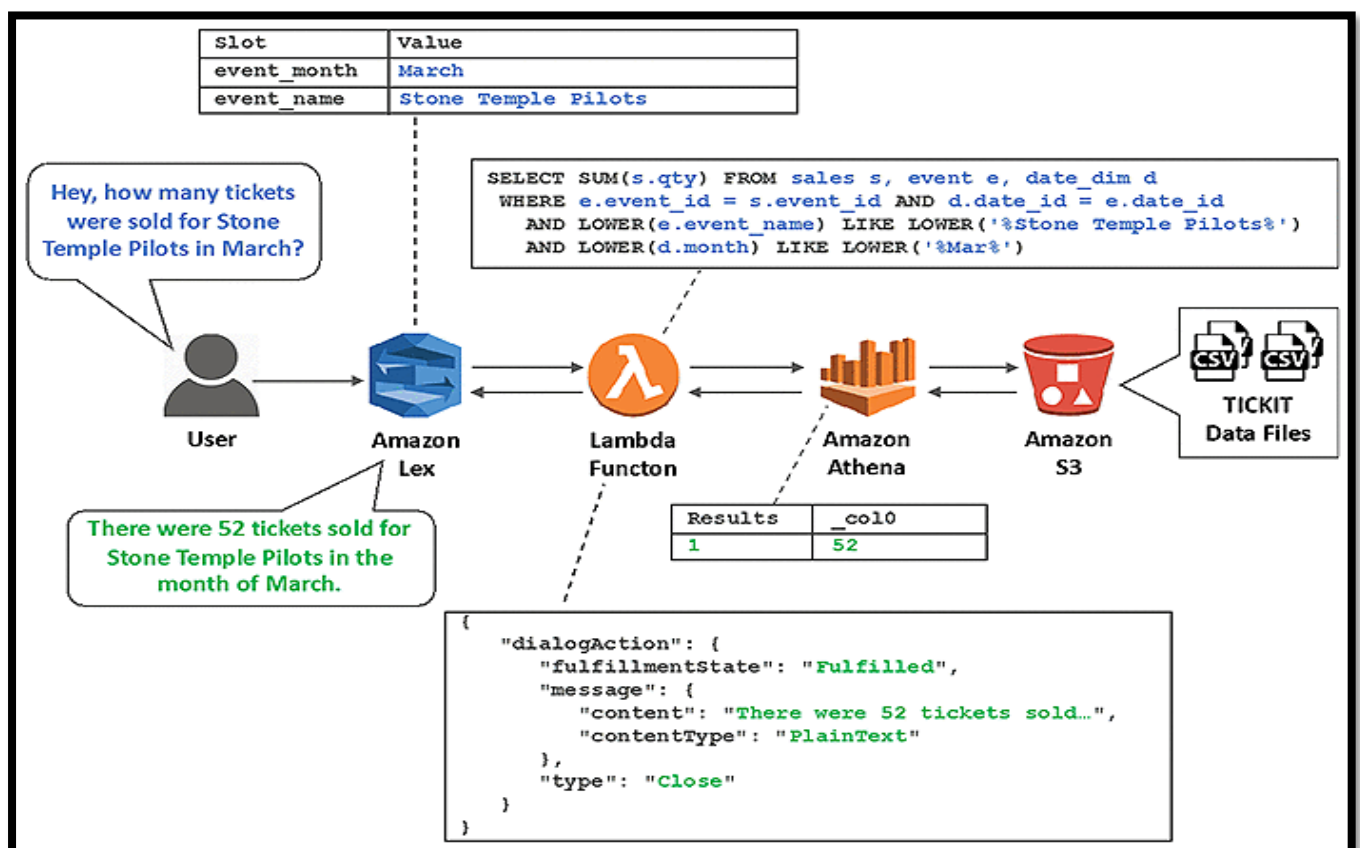
For the Relation to NLP using AWS lex: I read that, it uses the [Amazon Elasticsearch Service](#) (Amazon ES) and optionally [Amazon Kendra](#) to make your questions and answers searchable.

1. When a user asks a question, the Amazon ES powerful full-text search engine or Amazon Kendra's machine learning natural language search engine is used behind the scenes to find the answer that is the best match for that question.
2. respond to user questions about data in a database, by converting the questions into backend database queries, and transforming the result sets into natural language responses.

For example, the request “tell me the increase in inventory last month” could be translated to “select sum(item_qty) from inventory where month(received_date) = 10”.

Algorithm followed inside:

- A Lex bot directs each of the user’s questions to an intent, which parses the question into slots.
- The Amazon Lex bot then passes the intent and slot data to an AWS Lambda function, which uses the data to construct a SQL query, and execute it against an Amazon Athena database.
- Athena retrieves the query results from a set of CSV files stored in an Amazon S3 bucket and returns the result set back to the Lambda function, which converts it into a natural language response.



For designing the intent, lex supports eight intents: Hello, Top, Compare, count, Switch, Reset, Refresh, Goodbye. Then build the domain specific natural language processing.