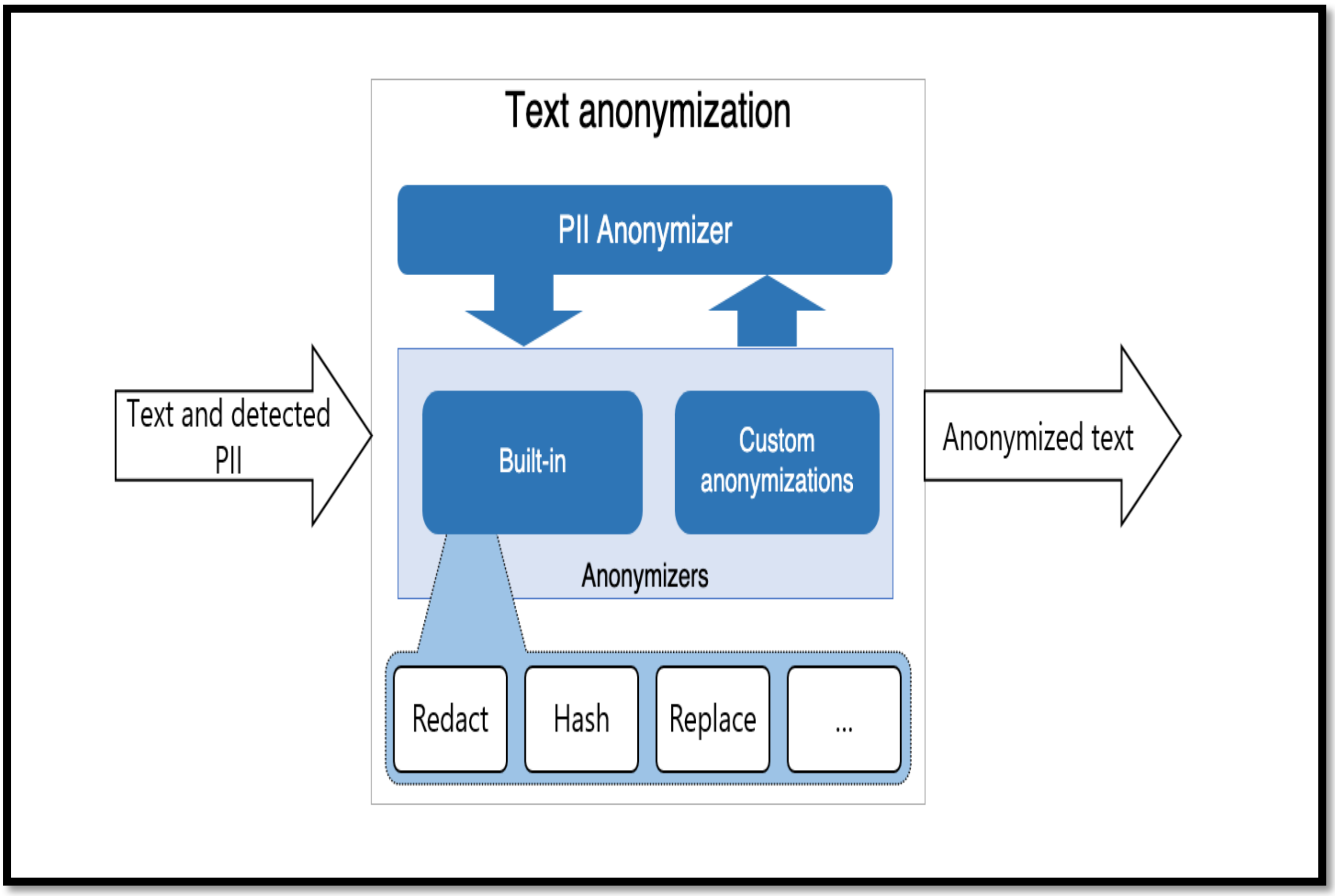
Problem statement:

Work on anonymizing using the Azure text analystics recognizer

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.

**

New thing:

Presidio analyzer is comprised of a set of PII recognizers which can run local or remotely. Azure Text Analytics is a cloud-based service that provides advanced natural language processing over raw text. One of its main functions includes Named Entity Recognition (NER), which can identify different entities in text and categorize them into pre-defined classes or types.

Text Analytics supports multiple PII entity categories. The Text Analytics service runs a predictive model to identify and categorize named entities from an input document

It uses various restful api by default such as

Spacy and swagger client and internally many more

Solved around many errors in importing and downloading Spacy, while swagger client is downloaded but it is showing error in installing, working on that …

New import statements to be added:

```
from presidio_analyzer import AnalyzerEngine

from text_analytics.example_text_analytics_recognizer import TextAnalyticsEntityCategory, TextAnalyticsRecognizer
```

then we would have defined the entities using textanalysis:

```
ta_entities = [
    TextAnalyticsEntityCategory(name="Medicine",
                                entity_type="NAME",
                                supported_languages=["en"]),
    TextAnalyticsEntityCategory(name="Age",
                                entity_type="AGE",
                                subcategory = "Age",
                                supported_languages=["en"]),
    TextAnalyticsEntityCategory(name="Disease",
                                entity_type="Disease",
                                supported_languages=["en"])]
```

from here we can instantiate and create our custom recognizer but the swager client api is not found in the module extracted.

Surfing through stackoverflow and data bricks documentations.

**

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**

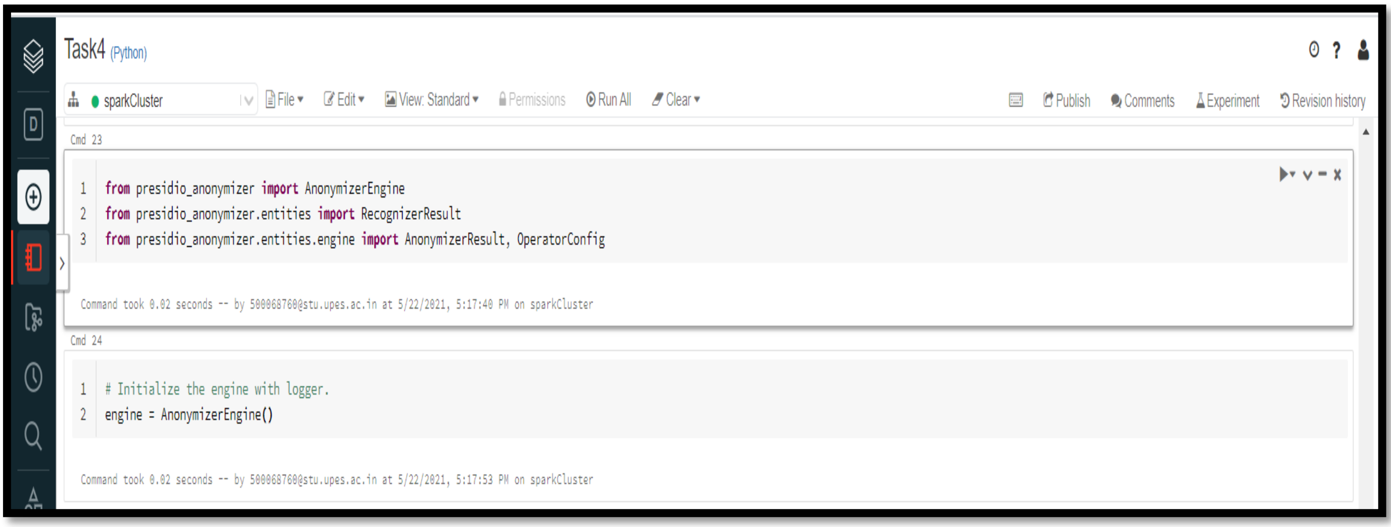
```
def anonymize_pii(text: str)-> str:
   # Anonymizers config to define the anonymization type.
   anonymized_data = engine.anonymize(
      text=text,
      analyzer_results=[RecognizerResult("MEDICINE", 18, 26, 0.8),
                    RecognizerResult("AGE", 41, 43, 0.8),
                    RecognizerResult("DISEASE", 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:



Cmd 23

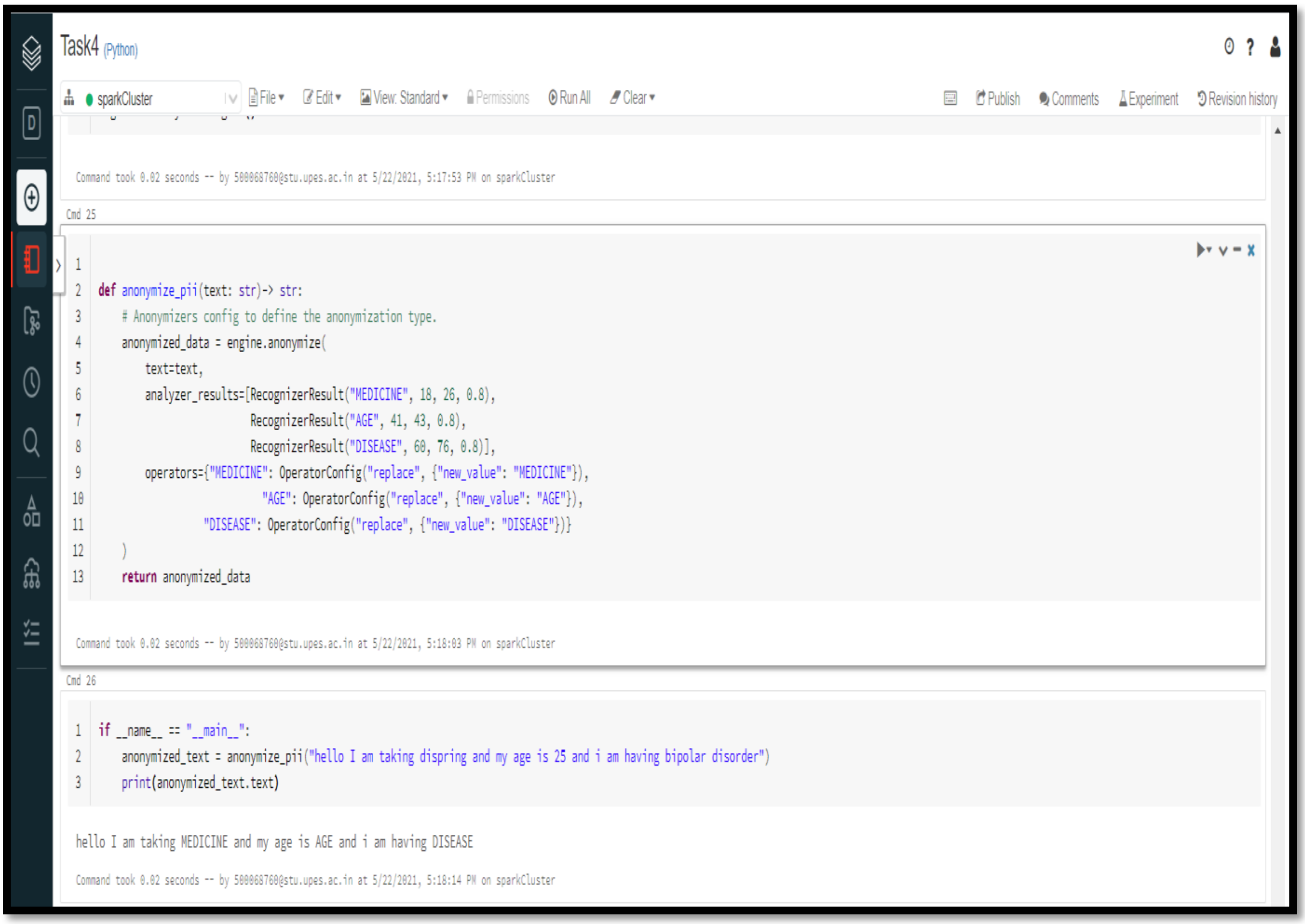
```python
from presidio_anonymizer import AnonymizerEngine
from presidio_anonymizer.entities import RecognizerResult
from presidio_anonymizer.entities.engine import AnonymizerResult, OperatorConfig
```

```
Command took 0.02 seconds -- by 500068760@stu.upes.ac.in at 5/22/2021, 5:17:40 PM on sparkCluster
```

Cmd 24

```python
# Initialize the engine with logger.
engine = AnonymizerEngine()
```

```
Command took 0.02 seconds -- by 500068760@stu.upes.ac.in at 5/22/2021, 5:17:53 PM on sparkCluster
```



```
Command took 0.02 seconds -- by 500068760@stu.upes.ac.in at 5/22/2021, 5:17:53 PM on sparkCluster
```

Cmd 25

```python

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
```

```
Command took 0.02 seconds -- by 500068760@stu.upes.ac.in at 5/22/2021, 5:18:03 PM on sparkCluster
```

Cmd 26

```python
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)
```

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

Command took 0.02 seconds -- by 500068760@stu.upes.ac.in at 5/22/2021, 5:18:14 PM on sparkCluster
```

How to related NLP with AWS Lex

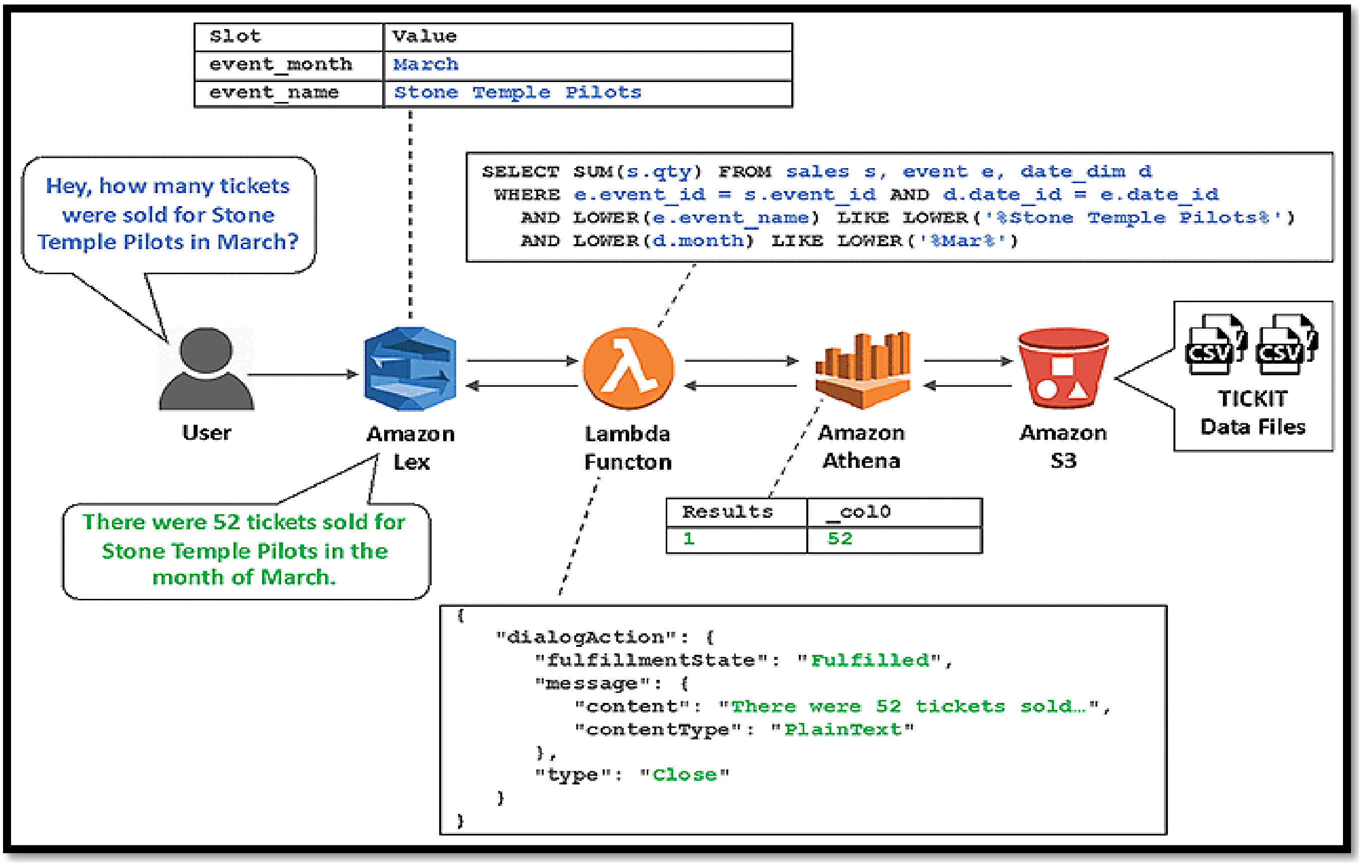
For the Relation to NLP using AWS lex: I read that, it uses the Amazon Elasticsearch Service (Amazon ES) and optionally Amazon Kendra to make your questions and answers searchable.

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.



Slot	Value
event_month	March
event_name	Stone Temple Pilots

```
SELECT SUM(s.qty) FROM sales s, event e, date_dim d
 WHERE e.event_id = s.event_id AND d.date_id = e.date_id
   AND LOWER(e.event_name) LIKE LOWER('%Stone Temple Pilots%')
   AND LOWER(d.month) LIKE LOWER('%Mar%')
```

Hey, how many tickets were sold for Stone Temple Pilots in March?
User
Amazon Lex
Lambda Functon
Amazon Athena
Amazon S3
CSV
CSV
TICKIT Data Files
There were 52 tickets sold for Stone Temple Pilots in the month of March.

Results	_col0
1	52

```
{
    "dialogAction": {
       "fulfillmentState": "Fulfilled",
       "message": {
          "content": "There were 52 tickets sold…",
          "contentType": "PlainText"
       },
       "type": "Close"
    }
}
```

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