JetFire

` **Start Date:** 01 November, 2019

INDEX

Business Objective –

Nowadays automation is the frontier of every business and IT industries. Artificial Intelligence/ NLP/ Machine Learning have invaded almost all sectors.

In our case , Airlines industry is smoothly working on NLP , for conversational purpose. You go to any airlines/ aviation website there will be a chat-bot ready to assist you 24*7 with your queries. However , our project requirement is a little erratic from the above one.

While booking a ticket, PNR is generated for each Fare basis and attached to it are the Fare rules generally defined by the GDS (Global Distribution System) or Airlines itself. Our objective is too automate the task of Technical Support guy sitting in back-end and manually initiating the Refund/ Cancellation procedure. If we can make our model to learn the fare Rules from the Corpus generated , we can think about automating the refund.

However there are pros and cons of this test case, but we are running POC for the sake of our satisfaction , that whether Airlines industry is vulnerable to automate the refund with help of NLP or not.

Scope of the Project-

- Research NLP/NLU frameworks (find difference between NLU/NLP) present
- Analyze the cancellation rules (50 rules) - problem definition
- Find the various patterns - understanding solution
- Test 3 frameworks (find the best engine) - test with 50 rules
- Define output required - this is in parallel
- Program for realtime i.e. send a rule, evaluate it in english, output the dataset, verify the dataset.
- Build maturity if new rules come thru how to contextualize those through human input.

Phase 1 – Maturity Identification of Engines and Frameworks

Step 1- Identify various NLP/NLU Engines

After a brief discussion about the project, the initial task was to identify which NLP engine is appropriate for our Project. We saw many Open source as well as commercial tools available, which can perform many tasks such as Tokenization, Intent classification, Document Extraction, Named Entity Recognition, Insult Detection, and many other features. Narrowing our requirements we saw 4 main NLP/NLU Engines , AWS Comprehend , RASA NLU , ApacheOpenNLP , DialogFlow, Snips NLU on 6 main factors –

- OpenSource/ Commercial
- Package/ Library or Tool
- Integration APIs
- Language Support
- Number of years been available
- Test Example

We saw how each NLP engine has different life cycle , and what are the common APIs available for integration.

What we observed –

1. Many APIs are common In all the NLP Engines. Provides integration for HTTP / REST / POST APIs
2. The main purpose of each NLP Engine is Intent classification and Entity Recognition
   (NLP Engines has set of predefined Entities which we will discuss in latter part of doc)

Step 2 – Identify various Python Frameworks

In NLP patterns can be trained In 2 ways:- Using Readymade ***NLP/NLU Engines*** and second is **Rule based Processing.** For Rule based processing, there are many Python Libraries and Frameworks developed over the time with various functionalities. We saw a couple of frameworks and went deep into some of them to train our patterns. In similar way we factored out each Framework on basis of 6 factors mentioned above.

For our understanding we studied:

- NLTK(Natural Language Toolkit)
- GENSIM,
- spaCy,
- coreNLP

Depending on the use case, maturity level and applications we figured out to demonstrate spaCy and CoreNLP.

Snippet of spaCy-

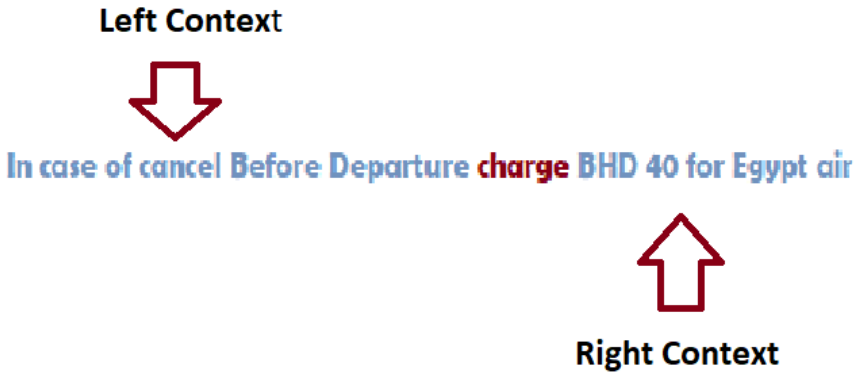
	SpaCy
Open Source/ Closed Source	It is an Open Source Library for advanced NLP written in C and Python. Allows for Cross platform operation
Library	The library provides extensive features as listed- • Non-destructive Tokenization • Named Entity Recognition (NER) • Pretrained vectors • POS tagging • Syntax driven Sentence Segmentation • Binary serialization
Integration API	Central data structures in SpaCy are – *doc* and *vocab*. Check the below link for Complete API reference:- https://spacy.io/api
Language Support	SpaCy provides support for 53 languages including common ones and many new ones. Some of the interesting new languages are listed below:- Greek, Lithuanian, Afrikaans, Bengali, Catalan, Telugu, Estonian, Kannada, Marathi, Latvian, Sinhala, Tagalog, Tatar, Urdu. Some of the language tokenizers require external dependencies- **Russian –** https://github.com/kmike/pymorphy2 **Ukranian –** https://github.com/kmike/pymorphy2 **Thai –** https://github.com/wannaphongcom/pythainlp **Chinese –** https://github.com/fxsjy/jieba **Japanese –** http://unidic.ninjal.ac.jp/back_number#unidic_cwj **Korean –** https://bitbucket.org/eunjeon/mecab-ko/src/master/README.md **Vietnamese –** https://github.com/trungtv/pyvi
Number of years	4 years [Initial release Febrary 2015]

One Framework we missed out was BERT –

BERT (Bidirectional Encoder Representation from Transformers) is a demystifying tool to the groundbreaking applications of NLP that are not covered in many other Libraries. Developed by Google, it is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.

- How BERT works:-

Suppose there are two sentences, and we want BERT to understand our language and derive a context out of it. he bidirectionality of a model is important for truly understanding the meaning of a language. Let's see an example to illustrate this. There are two context involved in this example and both of them are related to the word "charge" for our pattern:



If we try to predict the nature of the word “charge” by only taking either the left or the right context, then we will be making an error in at least one of the two given examples.

Here our Left context predicts – Cancellation Before Departure condition

Right context predicts – Currency of Refund (BHD) and Airlines Egypt Air)

One way to deal with this is to consider both the left and the right context before making a prediction. That’s exactly what BERT does!

Limitation –

BERT does not classify intent, prediction is just made on the context evaluation. Also entities are not recognized with this tool. However BERT stands to be the most powerful framework for Language understanding , Tokenization and Context Generation . The secret behind this framework is that it uses word2vector to BERT syntax and gives the predicted outcome.

Phase 2 – PNR Extraction , Fare Rules and Pattern Generation

Step 3 – Studying PNR rule -

After extracting 50 PNR rules from fare basis, we studied it and presented in a tabular way. Each rule depicted various functionalities such as

- Cancellation : Before Departure
  : After Depature
- No show
- Reissue/Revalidation
- Upgrade/ Downgrade

Example of generated ticket PNR

```
>FN*1/P16

001 BAHCAI 05JUN20 MS BHD
45.000 TRIMS STAY---/3M
BK-T

16. PENALTIES

UNLESS OTHERWISE SPECIFIED
NOTE - RULE 31ME IN IPRG

APPLIES

UNLESS OTHERWISE SPECIFIED

CANCELLATIONS

ANY TIME

CHARGE USD 50.00 FOR
CANCEL/REFUND.

WAIVED FOR DEATH OF
PASSENGER OR FAMILY MEMBER.

BEFORE DEPARTURE

CHARGE USD 75.00 FOR NO-
SHOW.

WAIVED FOR DEATH OF
PASSENGER OR FAMILY MEMBER.

)><

AFTER DEPARTURE

CHARGE 100 PERCENT FOR NO-
SHOW.

WAIVED FOR DEATH OF
PASSENGER OR FAMILY MEMBER.

CHANGES

ANY TIME

CHARGE USD 30.00 FOR
REISSUE/REVALIDATION.

WAIVED FOR DEATH OF
PASSENGER OR FAMILY MEMBER.
```

```
NOTE -

CHILD DISCOUNT DOES NOT
APPLY

INFANT NOT OCCUPYING A SEAT
FREE OF CHARGE

-----

IN CASE OF REISSUE/
REVALIDATION 24HOURS BEFORE

FLIGHT OR AFTER FLIGHT
DEPARTURE ONLY NOSHOW FEES

)><

TO BE APPLIED.

-----

UPGRADE TO HIGHER
COMPARTMENT ON SAME
FLIGHT/SAME

DATE IS PERMITTED WITHOUT A
CHARGE ONLY FARE

DIFFERENCE TO BE COLLECTED.

-----

REISSUE MUST BE COMPLETED 24
HOURS BEFORE FLIGHT

DEPARTURE OR WILL BE CHARGED
AS NOSHOW.

-----

FEE APPLIED PER CHANGED FARE
COMPONANT.

IF MULTIPLE CHANGES ARE MADE
AT THE SAME TIME

HIGHEST FEE WILL BE APPLIED
OF ALL CHANGED

FARE COMPONANTS.

)><
```

```
-----

REISSUE RESULTING FARE MUST
BE EQUAL OR HIGHER

THAN PREVIOUS BASIC FARE.

EXCEPTION FROM BUSINESS
CLASS TO Y RBD FARES IS

PERMITTED.

-----

IN CASE OF REISSUE OF UNUSED
TICKET APPLY CURRENT

FARES.

EXCEPTION FOR CHANGE OF ONLY
INBOUND FARE

COMPONENT APPLY HISTORICAL
FARE.

BOTH CHANGE FEES AND
DIFFERENCE IN FARE TO BE

COLLECTED.

-----

)><

BOTH CHANGE FEES AND
DIFFERENCE IN FARE TO BE

COLLECTED.

-----

IN CASE OF REISSUE OF
PARTIAL USED TICKETS APPLY

HISTORICAL FARES
RECALCULATED FROM THE LAST
FARE

BREAK POINT. ALL CONDITIONS
OF THE NEW FARES MUST

BE COMPLIED WITH BOTH CHANGE
FEES AND DIFFEREN
```

What we learned from the Fare rules : (Snpashot of above PNR)

BAH - CAI (BHD 50.00)

Function	Logical	Condition	Period	Exclude	Include	Amt/Percent	Value	CCY
Cancellation	NA	Any Time	-	-	-	Amount	50	USD
No Show	Before	Departure	-	-	-	Amount	75	USD
No Show	After	Departure	-	-	-	Percentage	100	-
ReIssue	-	Any Time	-	-	-	Amount	30	usd
ReIssue	-	Any Time	-	-	-	Amount	30	USD
ReIssue	Before	Departure	24 hours	-	-	-	75	USD
ReIssue	After	Departure	24 hours	-	-	-	75	USD
Discounts	-	-	-	CHD	ADT	-	-	-
Upgrade	-	Any Time	-	-	-	FREE	-	-

Observations:-

- For each Fare rules , the cancellation policy was different. Although after examining each rule we could find similarity in the rules and that lead us to create patterns
- All Flights inbound/Outbound Bahrain were charged in BHD
- Also many other factors such as Death waiver requirements, Upgrade/Downgrade conditions, Surcharge fees/ Service fees changes as per the PNR

Pattern Generation –

	A	B
1	Cancellation	In case of Cancel Anytime charge USD 50.00 with Death Waiver included for Egypt Air
2	Cancellation	In case of Refundable ticket , to cancel Anytime charge Penalty and refund BHD 16.00 for KLM Airlines
3	Cancellation	In case of Non Refundable ticket, no refund will be given for KLM Airlines
4	Cancellation	In case of cancel Before Departure charge BHD 20.00 with Death Waiver Included for Emirates
5	Cancellation	In case of cancel After Departure no refund will be given for Emirates
6	Cancellation	In case of cancel Before Departure charge USD 35.00 for Kuwait Airlines
7	Cancellation	In case of cancel After Departure if ticket is partially used apply general KAC Refund Policy with one way fare only for Kuwait Airlines
8	Cancellation	In case of cancel After Departure charge 75 percent of fare (BHD 60.00) or one way ticket fare for Kuwait Airlines
9	Cancellation	In case of cancel Before Departure Charge USD 80.00 with Medical waiver and Child Discount excluded for Turkish Airlines
10	Cancellation	In case of Cancel After Departure Charge Fare Difference between fare paid (BHD 61.00) and applicable fare for journey flown with respect to RBD code for Turkish Airlines
11	Cancellation	In case of cancel Before Departure Charge USD 80.00 with Medical waiver and Child Discount excluded for Turkish Airlines
12	Cancellation	In case of Cancel After Departure Charge Fare Difference between fare paid (BHD 68.00) and applicable fare for journey flown with respect to RBD code for Turkish Airlines
13	Cancellation	In case of cancel Anytime charge BHD 15.00 with Child discount Included for Cathay Pacific
14	Cancellation	In case of cancel Anytime no refund will be given with Death Waiver Included and Infant Discount Excluded for Oman Air
15	Cancellation	In case of Cancel Before Departure charge BHD 25.00 with Death Waiver included and Medical Waiver Excluded for Emirates

After creating patterns , we tried to create a Corpus dictionary to train our patterns using various Tools and libraries.

The main Intent of training the data was to parse the input rule pattern, identify the intent and Entities which we achieved through various tests.

{

INTENT : Cancellation

ENTITIES:

Condition

Currency

Airlines

Amount

Death Waiver Included

}

Phase 3 – Training Patterns, POC (Proof of Concept)

(1) AWS Comprehend

Identify Topics in collection of text , predict sentiment of text, Entity Recognition. Comprehend uses pre built Stanford's CoreNLP to classify NER.

We used Comprehend in 2 steps for our model

- Training the data set

  Go to Launch Comprehend -> Custom Named Entity Recongition -> Train recognizer -> Add Recognizer name -> Custom entity type -> Add location of S3 bucket where input csv file is to be trained -> Enter location of S3 bucket to store trained file -> Set IAM settings -> Check trained file

- Creating Analysis Job
  Go to create Job -> Name -> Analysis type -> Input file of trained data -> Output S3 bucket -> Access permission -> VPC settings -> Create Job

For our understanding we tried two possibilities in AWS comprehend – Custom Classification and Topic Modelling in Entity recognition .

I've uploaded the output of both the test cases on my AWS account. Please refer or it is added on the Trello board.

https://s3.console.aws.amazon.com/s3/buckets/fare-classifier/?region=us-east-1&tab=overview

Limitations –

The JSON output as well as csv file derived after creating Job where not giving same results as per our expectations. So we moved on to try other tools and libraries.

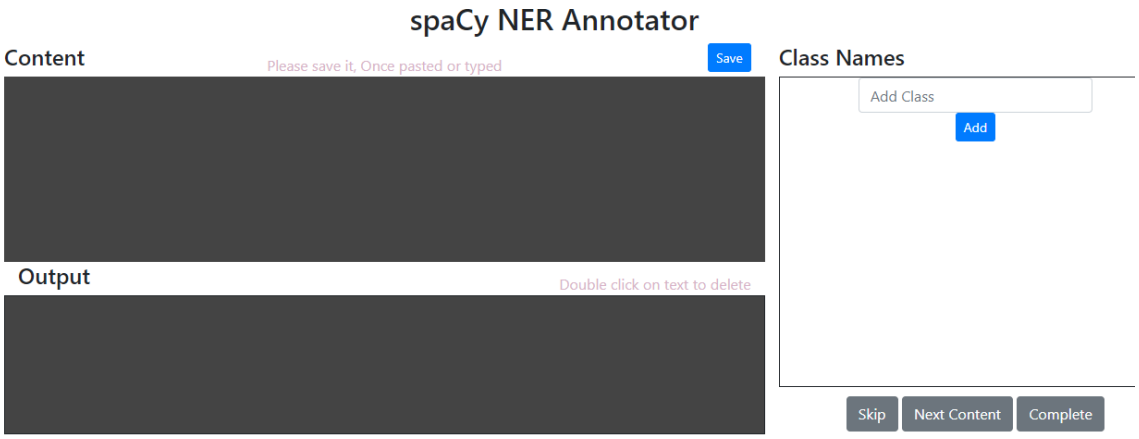
(2) spaCy

After discussing in brief about spaCy we saw various functionalities provided by spaCy.

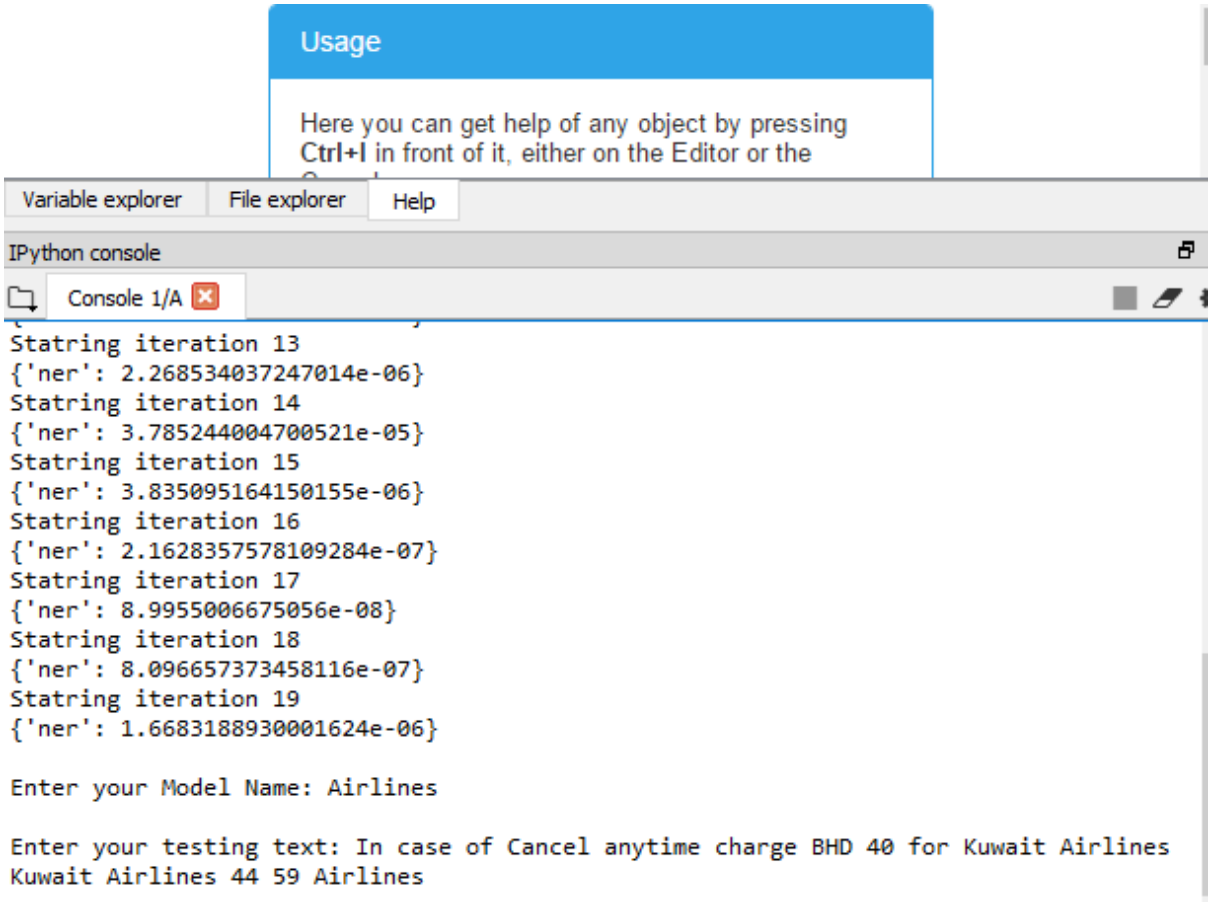
Our main purpose of using spaCy was to recognize Custom Entities , create our own dictionary of Airlines fare rules . We saw that in spaCy we can annotate our own labels using Prodigy . But after a little of research in spaCy Universe we saw "Spacy Annotator tool" which helped to train custom entities.

https://manivannanmurugavel.github.io/annotating-tool/spacy-ner-annotator/

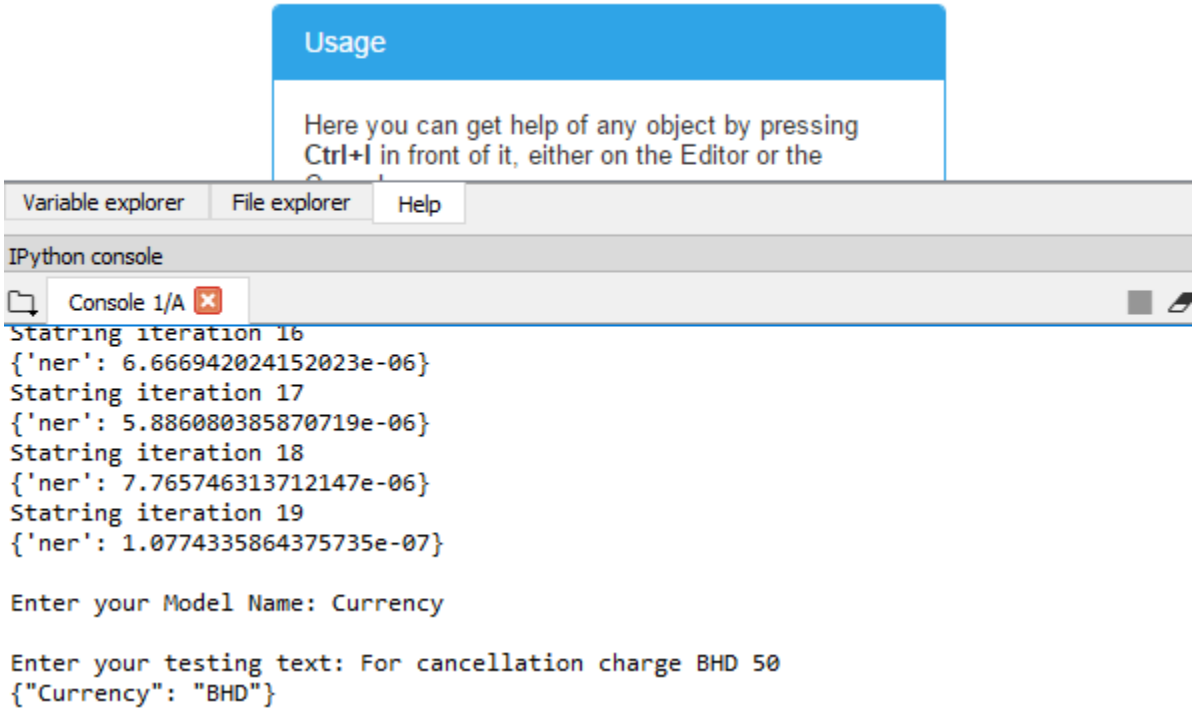
I tried pulling few patterns from our trained classifier and ran test cases. After training the rules the output was given in JSON format, and was used in Python script. I Created custom NER for recognizing two entities :- Currency and Airlines. The output of each test cases are shown below:



Trained NER for Airlines



Custom NER for Currency



What we learned:-

- Custom NER in Spacy was used to train multiple custom entities at a time but recognized only one entity
- The entities that were trained were only recognized. In order to gain precision we must add all the training patterns including all Currency and Airlines in a dictionary and then train them using the tool
- Also spaCy does not provide Intent Classification making it less efficient for us.

So we carried our next proof on Training patterns and Classifying intents using RASA

(3) RASA NLU

- A Libary for intent recognition and entity extraction based on SpaCy and Sklearn

NLP = NLU+NLG+ More

- NLP = understand,process,interprete everyday human language

- NLU = unstructured inputs and convert them into a structured form that a machine can understand and act upon

Uses

- Chatbot task
- NL understanding
- Intent classification

Process:-

Created a JSON file and Markdown file (Input) and trained them using RASA. Here spaCy was used in backend to load all the config files. I've uploaded .py script of RASA on AWS, please check

https://s3.console.aws.amazon.com/s3/buckets/rasa-demo/?region=us-east-1&tab=overview

Output:

#Prediction of Intent
Interpreter.parse(u" In case of cancellation charge BHD 50 for Egypt Air")

```
{'intent': {'name': 'cancelFlight', 'confidence': 0.7455215289019911},
 'entities': [{'start': 20,
   'end': 27,
   'value': 'BHD',
   'entity': 'currency',
   'confidence': 0.6636828413532201,
   'extractor': 'ner_crf'}
   'value': 'Egypt Air',
   'entity': Airlines
   'value': 50
   'entity': Money}],
 'intent_ranking': [{'name': 'cancelFlight',
   'confidence': 0.7455215289019911},
 'text': ' In case of cancellation charge BHD 50 for Egypt Air'}
```

Summary & Future Scope

After carrying out all the phases , we came to certain conclusion. I'm lisitng down the observation made by me as well as all of the Project members and what we can do in future to resume the POC:

1. As discussed in the objective, we were carrying a POC to check whether the patterns generated by us were compatible with frameworks and NLP tools available out there. We were pretty successful in carrying out many task such as NER, Intent Classification ,Pattern generation, Tokenization and Text Classification using various libraries

2. Each tool was independent of the function they were providing. We saw AWS to custom classify the labels (Cancellation/ No show) in JSON format as well as Topic modelling. SpaCy was used to train our own entities and recognize the. At the end RASA successfully gave us good result adding one more feature to the stack – Intent classification

3. Although after training phase, there were many limitations. The output obtained was not accurate/ usable to create a model. Also speaking about vulnerability , NLP is used in airlines industry only for conversational purpose. We cannot build a model to initiate refund involving customers' as well as company's Capital at stake until we have precision with our patterns

4. In future if we want to resume with the POC and work on model, we need maximum patterns to be trained and create our Neural network for Airlines Inventory using Deep learning.

References

I'm attaching few list of reference for future use, if anyone else wants to carry out the POC from where we have done.

1. https://spacy.io/
2. https://spacy.io/universe
3. https://spacy.io/api
4. https://github.com/Jcharis/Natural-Language-Processing-Tutorials/blob/master/Intent%20Classification%20With%20Rasa%20-%20Spacy/Intent%20Classification%20With%20Rasa%20NLU%20and%20SpaCy.ipynb
5. https://snips-nlu.readthedocs.io/en/latest/tutorial.html
6. https://docs.aws.amazon.com/comprehend/index.html
7. https://github.com/Jcharis/Natural-Language-Processing-Tutorials/tree/master/Intent%20Classification%20With%20Rasa%20-%20Spacy
8. https://github.com/chakki-works/doccano
9. https://manivannanmurugavel.github.io/annotating-tool/spacy-ner-annotator/
10. https://medium.com/@itsromiljain/build-a-conversational-chatbot-with-rasa-stack-and-python-rasa-nlu-b79dfbe59491
11. https://www.youtube.com/channel/UC2wMHF4HBkTMGLsvZAIWzRg
12. https://www.youtube.com/watch?v=sqDHBH9IjRU
13. https://www.youtube.com/watch?v=l4scwf8KeIA&t=995s