**Problem Statement**

An Indian startup named 'Foodie' wants to build a conversational bot (chatbot) which can help users discover restaurants across several Indian cities. You have been hired as the lead data scientist for creating this product.

The main purpose of the bot is to help users discover restaurants quickly and efficiently and to provide a good restaurant discovery experience. The project brief provided to you is as follows.

The bot takes the following inputs from the user:

* **City:** Take the input from the customer as a text field. For example:

|  |
| --- |
| Bot: In which city are you looking for restaurants?  User: anywhere in Delhi |



**Important Notes:**

* Assume that Foodie works **only in Tier-1 and Tier-2 cities**. You can use the [current HRA classification of the cities from here](http://en.wikipedia.org/wiki/Classification_of_Indian_cities). Under the section 'current classification' on this page, the table categorizes cities as X, Y and Z. Consider 'X ' cities as tier-1 and 'Y' as tier-2.
* The bot should be able to **identify common synonyms** of city names, such as Bangalore/Bengaluru, Mumbai/Bombay etc.

Your chatbot should provide results for tier-1 and tier-2 cities only, while for tier-3 cities, it should reply back with something like "We do not operate in that area yet".

* **Cuisine Preference:** Take the cuisine preference from the customer. The bot should list out the following **six cuisine categories** (Chinese, Mexican, Italian, American, South Indian & North Indian) and the customer can select any one out of that. Following is an example for the same:

|  |
| --- |
| Bot: What kind of cuisine would you prefer?   * + Chinese   + Mexican   + Italian   + American   + South Indian   + North Indian   User: I’ll prefer Italian! |

* **Average budget for two people:** Segment the price range (average budget for two people) into **three price categories**: lesser than 300, 300 to 700 and more than 700. The bot should ask the user to select one of the three price categories. For example:

|  |
| --- |
| Bot: What price range are you looking at?   * + Lesser than Rs. 300   + Rs. 300 to 700   + More than 700   User: in range of 300 to 700 |



While showing the results to the user, the bot should display the **top 5 restaurants** in a **sorted order** (descending) of the average Zomato user rating (on a scale of 1-5, 5 being the highest). The **format should be**: {restaurant\_name} in {restaurant\_address} has been rated {rating}.

Finally, the bot should ask the user whether he/she wants the details of the **top 10 restaurants on email**. If the user replies 'yes', the bot should **ask for user’s email id** and then send it over email. Else, just reply with a 'goodbye' message. The mail should have the following details for each restaurant:

* Restaurant Name
* Restaurant locality address
* Average budget for two people
* Zomato user rating

You can refer to the following links on how to send emails through Python:

* [Python Email Package](https://docs.python.org/3/library/email.examples.html)
* [Python Flask Mail](https://pythonprogramming.net/flask-email-tutorial/)

(A heads up: You'll have to enable secure access on Gmail to allow Python to send emails).

You have been given some sample conversational stories in the ‘test’ file. Look at the stories and try to observe entities, intents, dialogue-flow which may be useful to train the NLU and also the dialogue flow.

**[Sample conversational stories](https://cdn.upgrad.com/UpGrad/temp/72761812-7427-4f01-9343-90b1ae434914/Conversational_stories.pdf" \o "Conversational_stories.pdf" \t "_blank)**

File is provided

**Goals of this Project**

In this project, you will build a chatbot for ‘Foodie’ and then deploy it on Slack. The folder with starter codes has already been shared with you in session-1. You need to accomplish the following in the project:

1. **NLU training**: You can use rasa-nlu-trainer to create more training examples for entities and intents. Try using regex features and synonyms for extracting entities.
2. **Build actions for the bot**. Read through the Zomato API documentation to extract the features such as the average price for two people and restaurant’s user rating. You also need to build an ‘action’ for sending emails from Python.
3. **Creating more stories:** Use train\_online.py file to create more stories. Refer to the sample conversational stories provided above.  Your bot will be evaluated on something similar to the test stories shared.
4. **Deployment (Optional):** Deploy the model on Slack. You can create a new workspace or deploy it on an existing workspace (if you already use Slack).