**NLP UI to Predict Accident Severity Level**

Contents

[**1.** **NLP UI Description** 2](#_Toc70252063)

[**1.1.** **Milestone-1 Tasks** 4](#_Toc70252064)

[**1.2.** **Milestone-2 Tasks** 5](#_Toc70252065)

[**2.** **Appendix** 7](#_Toc70252066)

[**2.1** **UI Execution Screenshots from v1** 7](#_Toc70252067)

[**2.2** **UI Execution Screenshots from v2** 10](#_Toc70252068)

# **NLP UI Description**

This section of the document will walk you through the steps of using NLP User Interface to predict Accident Level Severity. The UI is built on Flask framework.

**File Location:**

<https://github.com/APR20CapstoneNLPGroup/chatbot/tree/main/Milestone%203/Final>

The folder structure is as below:

**Main folder/**

**|**

**|\_\_templates/**

**|**

**|\_prediction.html**

**|\_search.html**

**|\_models/**

**|\_ Final\_NLP\_UI\_v1.ipynb**

**|\_ Final\_NLP\_UI\_v2.ipynb**

**|\_USE\_LSTM.ipynb**

**|\_USE\_LSTMwATTN.ipynb**

**|\_ preprocess\_train\_v1.py >** (##Validation is not oversampled)

**|\_ preprocess\_train\_v2.py >** (##Validation is oversampled)

**|\_w2v\_model\_dir/**

**|**

**|\_variables/**

**|**

**|\_assets/**

**|**

**|\_saved\_model.pb**

**|\_[UI\_Screenshot](https://github.com/APR20CapstoneNLPGroup/chatbot/tree/main/Milestone%203/Final/UI_Screenshot" \o "UI_Screenshot)/**

**|**

**|\_**[**UI\_Snap.PNG**](https://github.com/APR20CapstoneNLPGroup/chatbot/blob/main/Milestone%203/Final/UI_Screenshot/UI_Snap.PNG)

**|\_data.csv**

**|\_IHMStefanini\_industrial\_safety\_and\_health\_database\_with\_accidents\_description.csv**

**|\_**[**NLP chatbot model report\_v2.xlsx**](https://github.com/APR20CapstoneNLPGroup/chatbot/blob/main/Milestone%203/Final/NLP%20chatbot%20model%20report_v2.xlsx)

**|\_Readme.txt**

**|\_model\_lstm.h5**

The **Final\_NLP\_UI\_v1.ipynb** is the main API file, which includes Flask code, callable modules to Load, Import, Clean, Pre-Process, Train/Test Split, Build Model, Train Model and finally Predict.

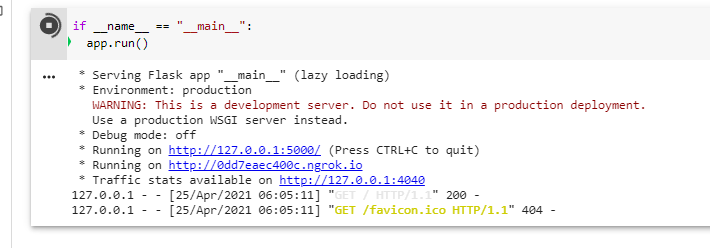
All the functions are consolidated in python (.py) file: **preprocess\_train\_v1.py**

**Step 1) Run Final\_NLP\_UI\_v1.ipynb**

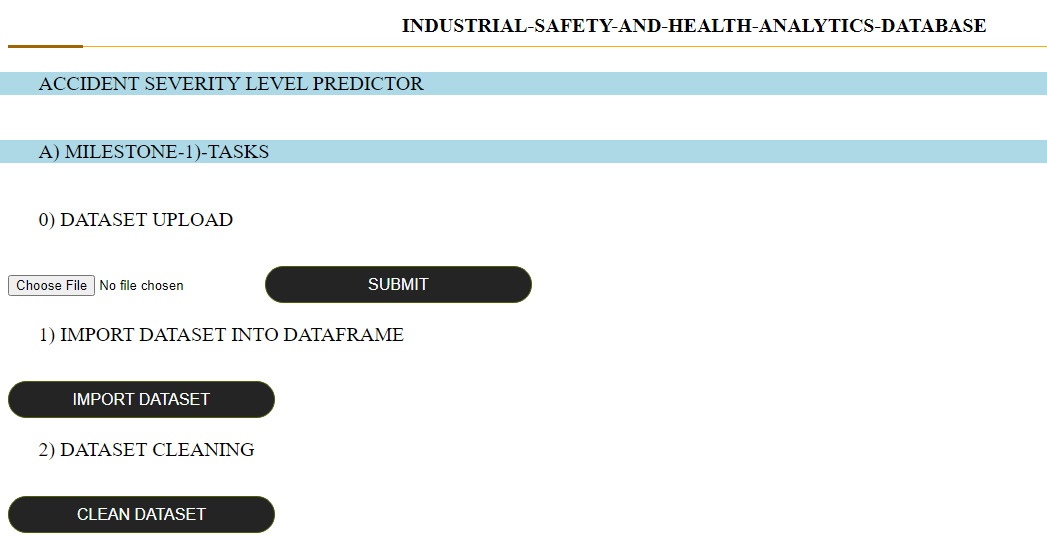
Towards the end, you will see “Running on [http://0dd7eaec400c.ngrok.io](http://0dd7eaec400c.ngrok.io/)” or similar to this.

We have utilized package **ngrok** to all Flask server be accessible over Internet.

Click on the ngrok provided link: [http://0dd7eaec400c.ngrok.io](http://0dd7eaec400c.ngrok.io/)



A new tab opens up as shown below:



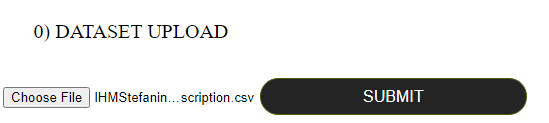
## **Milestone-1 Tasks**

Under **Milestone-1**, we have following tasks on UI interface:

1. Dataset Upload
2. Import Dataset into Dataframe
3. Clean Dataset

**Step 2) Choose File**

**Input**



Select the dataset and click “SUBMIT”. This function will upload the file in the same directory where main file **Final\_NLP\_UI\_v1.ipynb** is saved. The success will give following message:



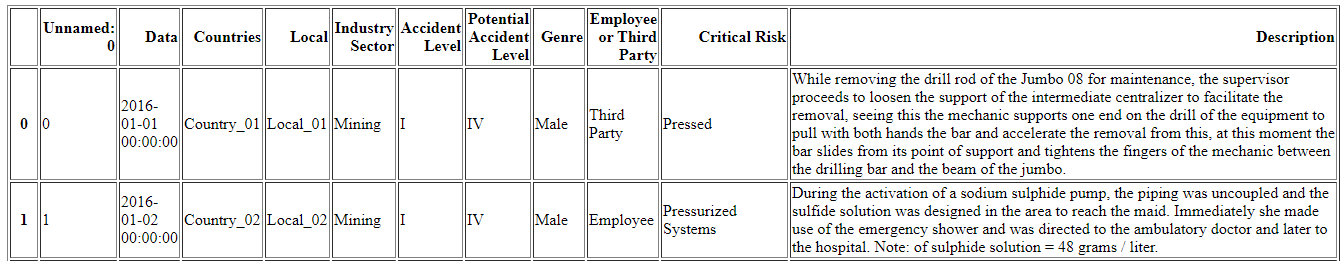
**Step 3) Import the Dataset into DataFrame**

**Input:**



Click on “IMPORT DATASET” and it will import the uploaded dataset into dataframe format. This dataset is still raw. This function will open a new tab to show the dataset in dataframe format shown below:

**Response:**



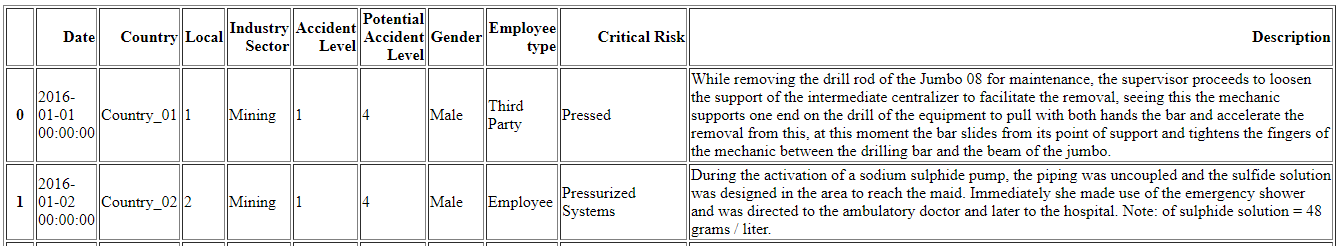
**Step 4) DataSet Cleaning**



By clicking this button, it will perform following clean operations:

* Drop Unnamed column
* Rename the column names {'Data':'Date', 'Countries':'Country', 'Genre':'Gender', 'Employee or Third Party':'Employee type'}, inplace=True)
* Replace values in columns:
  + Country
  + Local
  + Accident Level
  + Potential Accident Level
* Finally pickle dump the cleaned dataset: pickle.dump( data, open( "clean\_data.p", "wb" ) )

**Response:**



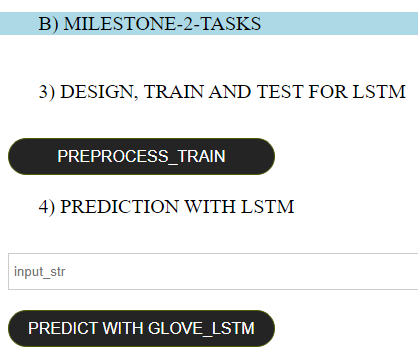
**This marks completion of Milestone 1 tasks**

Next, we move on to Milestone-2 tasks

## **Milestone-2 Tasks**

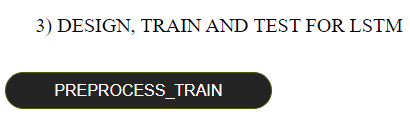
Under **Milestone-2**, we have following tasks on UI interface:

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Functions** | **UI Button** |
| **DESIGN, TRAIN AND TEST FOR LSTM** | Pre-Process with NLP  Train/Test Split  Model Build  Model Compile  Model Train  Model Fit | PreProcess\_Train |
| **Prediction** | Prediction | Predict with Glove\_LSTM |



**Step 5) Click PreProcess\_Train**

Upon click this button, it will run all the above tasks as listed in the table above.



Post completing, it will display following success message:

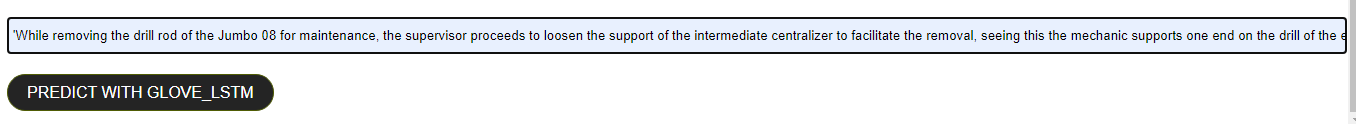
**Response:**



**Step 6) Prediction: Predict with Glove\_LSTM**

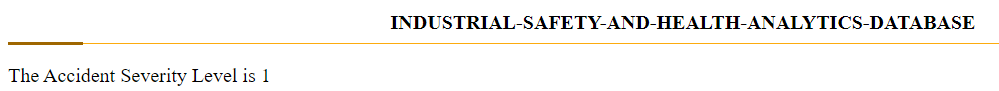
Here, please enter the Description of the Accident in the raw format, by tying into the ***Input Box***  as shown below:

**Query:**



Then click **Predict with Glove\_LSTM.**  This will display the Accident Severity Level as below:

**Response:**



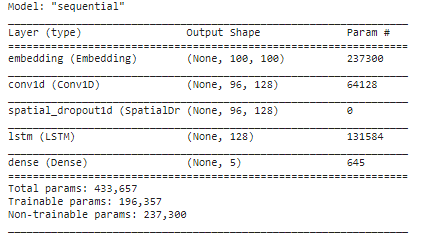
For the above provided Accident description, the Accident severity is Level 1. This is correct prediction.

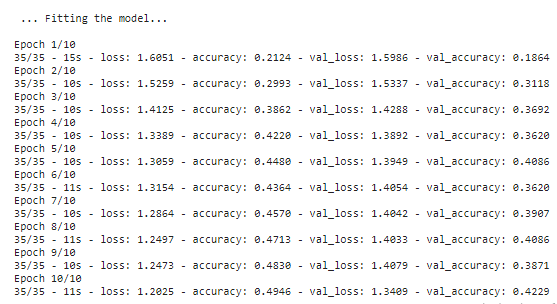
One can enter more accident descriptions and query the model to get the accident severity level.

**This completes the Milestone-2 tasks.**

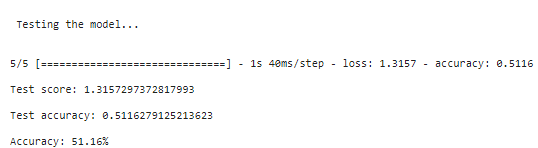
# **Appendix**

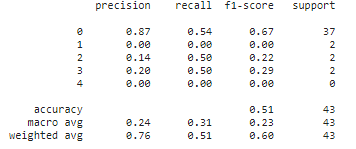
## **UI Execution Screenshots from v1**

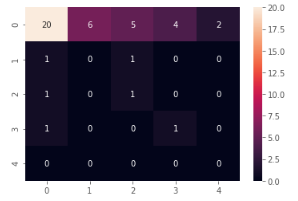


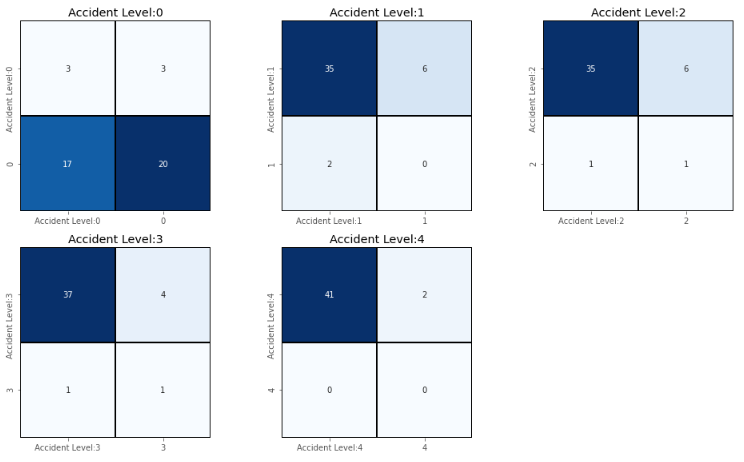
















## **UI Execution Screenshots from v2**

