**INTERNSHIP: PROJECT REPORT**

------------------------------------------------------------------------------------------------------------------------------------------

|  |  |
| --- | --- |
| Internship Project Title | Human Emotion Sentimental Analysis |
| Project Title | RIO-45 Automate detection of different emotions from textual comments and feedback |
| Name of the Company | TCS ION |
| Name of the Industry Mentor | Mr. Debashis Roy |
| Name of the Institute | Prepinsta Technologies PVT LTD |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 30/11/21 | 19/12/21 | 7 | VS Code  Google Chrome  Kaggle | Python 3.9 |
| **Project Synopsis:**  The field of study that focuses on the interactions between human language and computers is called natural language processing, or NLP for short. It sits at the intersection of computer science, artificial intelligence, and computational linguistics.  Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation. | | | | |
| **Solution Approach:**  In this project we are going to apply Deep Learning Algorithm such as Recurrent Neural Network (RNN). First i take a dataset from kaggle for this project which is Human Emotion Sentimental Analysis. The dataset contain 20000 data. From the dataset first i remove the stop words using NLTK library and then convert the sentiments ('sadness', 'anger', 'love', 'surprise', 'fear', 'joy') into numerical values. Thereafter i need to convert the words in to a sequence of array which is also called “Bag of Words”. Then i create a model using deep learning and predict with respect to some random sentence. | | | | |
| **Assumptions:**  The dataset has 20000 amounts of data from the Kaggle. The dataset first split 18000 data for training and 2000 data for testing. Then I convert the sentiments into numerical values and convert the numerical into categorical value. In the model 0 means “anger” , 1 means “fear”, 2 means “joy”, 3 means “love”, 4 means “sadness” and 5 means “surprise”.  **Project Diagrams:**  **Layer Model Flow:**      **Algorithm :**  Here in this project I used a RNN (Recurrent Neural Network) using normal Dense Layers. Actually I used 3 main Dense Layers and 1 final layer. | | | | |
| **Outcome:**  **Prediction1:**   * **Statement: “**i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake**”** * **Output:** sadness * **Actual:** sadness   **Prediction2:**   * **Statement: “**im grabbing a minute to post i feel greedy wrong” * **Output:** anger * **Actual:** anger   **Prediction3:**  **Statement: “**i do not feel reassured anxiety is on each side**”**   * **Output:** joy * **Actual:** joy   **Prediction4:**  **Statement: “**i feel romantic too**”**   * **Output:** love * **Actual:** love   **Prediction5:**   * **Statement: “**i am now nearly finished the week detox and i feel amazing” * **Output:** surprise * **Actual:** surprise   **Prediction6:**   * **Statement: “**i had stated to her the reason i feel so fearful is because i feel unsafe**”** * **Output:** fear * **Actual:** fear | | | | |
| **Model Definition:**  In this model I used 4 layers, 3 Dense layers and 1 final layer and the Activation Function is Relu.  **Compile Model:**  The number of iteration is 20 i.e. epochs=20.  Epoch 1/20  282/282 [==============================] - 7s 14ms/step - loss: 0.7989 - accuracy: 0.7177 - val\_loss: 0.4255 - val\_accuracy: 0.8555  Epoch 2/20  282/282 [==============================] - 4s 13ms/step - loss: 0.2191 - accuracy: 0.9218 - val\_loss: 0.4265 - val\_accuracy: 0.8635  Epoch 3/20  282/282 [==============================] - 3s 12ms/step - loss: 0.0994 - accuracy: 0.9658 - val\_loss: 0.5249 - val\_accuracy: 0.8590  Epoch 4/20  282/282 [==============================] - 4s 12ms/step - loss: 0.0538 - accuracy: 0.9810 - val\_loss: 0.6456 - val\_accuracy: 0.8440  Epoch 5/20  282/282 [==============================] - 4s 12ms/step - loss: 0.0332 - accuracy: 0.9883 - val\_loss: 0.7150 - val\_accuracy: 0.8380  Epoch 6/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0271 - accuracy: 0.9906 - val\_loss: 0.7642 - val\_accuracy: 0.8485  Epoch 7/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0203 - accuracy: 0.9928 - val\_loss: 0.8370 - val\_accuracy: 0.8465  Epoch 8/20  282/282 [==============================] - 3s 12ms/step - loss: 0.0158 - accuracy: 0.9941 - val\_loss: 0.9387 - val\_accuracy: 0.8455  Epoch 9/20  282/282 [==============================] - 3s 12ms/step - loss: 0.0182 - accuracy: 0.9931 - val\_loss: 0.9719 - val\_accuracy: 0.8395  Epoch 10/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0245 - accuracy: 0.9907 - val\_loss: 0.9553 - val\_accuracy: 0.8370  Epoch 11/20  282/282 [==============================] - 5s 16ms/step - loss: 0.0195 - accuracy: 0.9919 - val\_loss: 1.0053 - val\_accuracy: 0.8315  Epoch 12/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0275 - accuracy: 0.9889 - val\_loss: 1.0614 - val\_accuracy: 0.8410  Epoch 13/20  282/282 [==============================] - 5s 16ms/step - loss: 0.0213 - accuracy: 0.9908 - val\_loss: 1.0304 - val\_accuracy: 0.8435  Epoch 14/20  282/282 [==============================] - 4s 14ms/step - loss: 0.0137 - accuracy: 0.9941 - val\_loss: 1.0721 - val\_accuracy: 0.8375  Epoch 15/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0096 - accuracy: 0.9954 - val\_loss: 1.1372 - val\_accuracy: 0.8430  Epoch 16/20  282/282 [==============================] - 3s 12ms/step - loss: 0.0076 - accuracy: 0.9962 - val\_loss: 1.1588 - val\_accuracy: 0.8435  Epoch 17/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0073 - accuracy: 0.9958 - val\_loss: 1.1827 - val\_accuracy: 0.8380  Epoch 18/20  282/282 [==============================] - 4s 13ms/step - loss: 0.0067 - accuracy: 0.9958 - val\_loss: 1.1690 - val\_accuracy: 0.8420  Epoch 19/20  282/282 [==============================] - 4s 14ms/step - loss: 0.0066 - accuracy: 0.9959 - val\_loss: 1.2129 - val\_accuracy: 0.8460  Epoch 20/20  282/282 [==============================] - 5s 17ms/step - loss: 0.0065 - accuracy: 0.9961 - val\_loss: 1.1825 - val\_accuracy: 0.8415  **For Training Loss: 0.0065**  **For Training Accuracy: 99.61**  **For Testing Loss:1.1825**  **For Testing Accuracy: 84.15**  **Exceptions considered:**  This model is for short sentiment. If the sentiment is long it may not predict right prediction. But for short sentiment it remain same. | | | | |
| **Enhancement Scope:**  Increasing the accuracy score is the biggest enhancement scope. It may increase after increasing number of layers in the model, but the result may also reverse or remain same. | | | | |
| **Link to Code and executable file:**  **GitHub Link:** [**https://github.com/Debanjan200/Tcs\_Ion\_Internship**](https://github.com/Debanjan200/Tcs_Ion_Internship) | | | | |