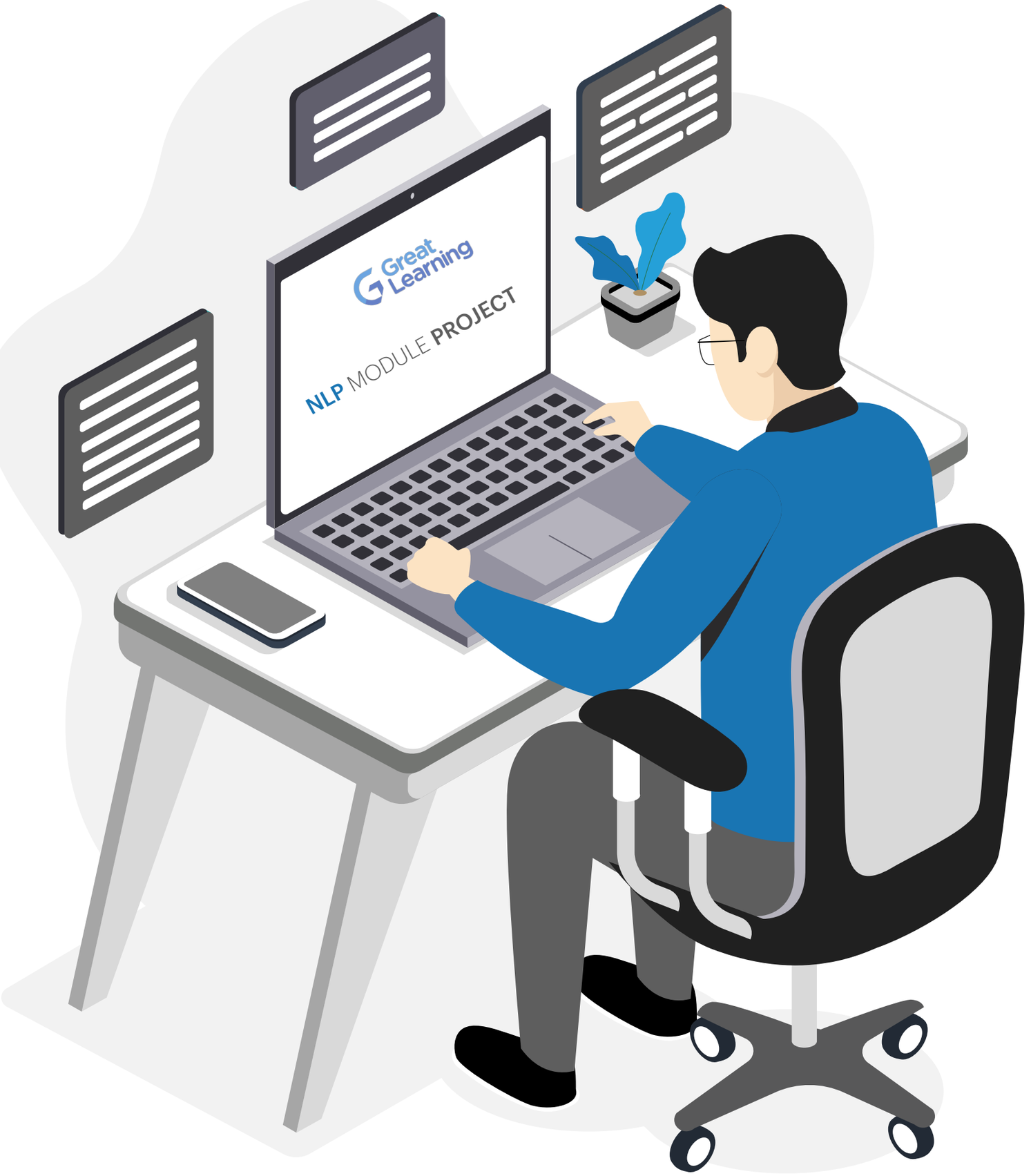

AIFL | MODULE PROJECT

©Great Learning. Proprietary content. All Rights Reserved. Unauthorised use or distribution prohibited

Natural Language Processing

TOTAL SCORE 30

Part A - 15 Marks

- **DOMAIN:** Social media analytics

- **CONTEXT:** Past studies in Sarcasm Detection mostly make use of Twitter datasets collected using hashtag based supervision but such datasets are noisy in terms of labels and language. Furthermore, many tweets are replies to other tweets and detecting sarcasm in these requires the availability of contextual tweets.In this hands-on project, the goal is to build a model to detect whether a sentence is sarcastic or not, using NLP techniques.

- **DATA DESCRIPTION:** The dataset is collected from two news websites, theonion.com and huffingtonpost.com.
  - This new dataset has the following advantages over the existing Twitter datasets:
  - Since news headlines are written by professionals in a formal manner, there are no spelling mistakes and informal usage. This reduces the sparsity and also increases the chance of finding pre-trained embeddings.
  - Furthermore, since the sole purpose of The Onion is to publish sarcastic news, we get high-quality labels with much less noise as compared to Twitter datasets.
  - Unlike tweets that reply to other tweets, the news headlines obtained are self-contained. This would help us in teasing apart the real sarcastic elements
  - Content: Each record consists of three attributes:
    - is_sarcastic: 1 if the record is sarcastic otherwise 0
    - headline: the headline of the news article
    - article_link: link to the original news article. Useful in collecting supplementary data
  - Reference: https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection

- **OBJECTIVE:**

  Build a sequential NLP classifier which can use input text parameters to determine the customer sentiments.

- **STEPS AND TASKS:**
  1. Read and explore the data [2 Marks]

     [Hint: Use data explorer block to analyse the entire dataset]
  2. Transform strings to document [2 Marks]
  3. Preprocessing [8 Marks]

     [Hint: Apply all the pre-processing techniques learnt during the course like punctuation remover, filtering numbers, stop word filter, character filter, case converter, stemming/lemmatization, Bag of words(BOW), Term Frequency(TF)]
  4. Create bit vectors for documents [2 Marks]
  5. Extract sentiment label [1 Marks]
  6. Partition the dataset [1 Marks]
  7. Model building [8 Marks]

     [Hint: Apply at least four models that you learnt during the course and comment your findings]
  8. Hyper parameter tuning [4 Marks]

     [Hint: Fine tune all the model parameters, including varying the different pre-processing techniques and share the best model with classification metrics]
  9. Conclusion [2 Marks]

     [Hint: Comment on the findings that you inferred during the entire course of the project]

***Note:**

1. *As the dataset is heavy the model takes more time to run. You can either reduce the dataset size or use Java Edit variable tool box which can help you run with important terms in the document. You can refer to the link on how to use the tool box (https://www.knime.com/blog/sentiment-analysis).*
2. *Regarding model performance, if the accuracy is greater than 60% then then model is performing good for the given dataset.*

©Great Learning. Proprietary content. All Rights Reserved. Unauthorised use or distribution prohibited