Text documents are essential as they are one of the richest sources of data for businesses. Text documents often contain crucial information which might shape the market trends or influence the investment flows. Therefore, companies often hire analysts to monitor the trend via articles posted online, tweets on social media platforms such as Twitter or articles from newspaper. However, some companies may wish to only focus on articles related to technologies and politics. Thus, filtering of the articles into different categories is required.

Often the categorization of the articles is conduced manually and retrospectively; thus, causing the waste of time and resources due to this arduous task. **Hence, your job as a machine learning engineer is tasked to categorize unseen articles into 5 categories namely Sport, Tech, Business, Entertainment and Politics.**

Data can be obtained from

https://raw.githubusercontent.com/susanli2016/PyCon-Canada-2019-NLP-Tutorial/master/bbc-text.csv by simply passing this URL into *pd.read_csv(URL)*

The following are the criteria of your project:

1) Develop your own model using LSTM which can achieve accuracy of more than 70% and F1 score of more than 0.7.
2) You are only allowed to use TensorFlow library to develop and train the model.
3) Plot the graph using Tensorboard.
4) Save the model in *.h5* format in a folder named saved_models.
5) Save tokenizer in *.json* format in a folder named saved_models

Tip: you may train and display tensorboard using Google Colab then download the trained model after training and screenshot respectively.

Files to be submitted and uploaded to GitHub and LMS (submission link will be given on the assessment day):

1) Training and a script containing the module (GitHub and LMS)
2) Saved model in .h5 format and scalers (if any) in .pkl file format. (GitHub and LMS)
3) Training process plotted using Tensorboard can be snipped and saved as image file format (LMS).

4) A screenshot of your developed model's architecture should be saved as .png file format and zipped in a folder together with the rest of the files for LMS submission. Also include the screenshot in README.md and display on your GitHub repo (GitHub and LMS)
5) Performance of the model and the reports can be snipped and saved as image file to be included in the zip folder for LMS submission. (LMS and GitHub)
6) Include your GitHub URL directing to your assessment 2 in a text file then submit to LMS. (LMS)
7) **Don't forget to credit/cite the source of the data on your GitHub page.**

*Please zip all the required files into one folder then submit to LMS.

**Please save the dataset and model in 2 different folders to GitHub.

Complete the assessment and submit the files to LMS and GitHub by 5pm. Good Luck!!!

	100%	50%	0%
Task Completion (30%)	Scripts can be executed without any error on trainer's local machine.	-	Scripts fail to be executed on trainer's local machine.
Project requirements (30%)	Able to achieve the objectives of the project using relevant and appropriate approach.	Able to achieve the objectives of the project but using inappropriate approach such as brute forcing the solution.	Fail to achieve the objectives of the project.
Exploratory data analysis (30%)	Demonstrates strong understanding on the objectives of the project and performs relevant approach to process the data. Necessary data processing techniques such as, data loading, data cleaning, features selection and data preprocessing are performed and well justified.	Shows comprehensive understanding of the objectives of the project but uses incorrect or irrelevant approach to process the data. For example, removing NaN data when there is limited amount of samples in the dataset.	Shows limited understanding of the objectives of the project. Absence of data processing section in the code.
Code readability (5%)	Involves the usage of functions or methods for repeated tasks. Codes are easily readable and justified by including comments and description texts.	Minimal usage of functions or methods for repeated tasks. Available comments and descriptions but lack of details.	No usage of functions or methods for repeated tasks. Codes are difficult to read and understand. Missing descriptions and comments.
GitHub repo (4%)	Detailed and clear instructions of the project on README.md. Results such as graphs are also included in README.md as part of the project description.	Project successfully uploaded to GitHub repo but with incomplete README.md. Missing descriptions, instructions, and results.	Fails to upload project to GitHub repo and missing README.md
PEP8 compliance (1%)	Fully complies with PEP 8 Standard	Partially complies with PEP 8 Standard	Fails to comply with PEP 8 Standard
Total (100%)			