**Project-Exam 2 Deadline on 29 July 2020 11:59 PM**

1. Implement text classification on the [review’s sentiment dataset](https://umkc.box.com/s/nfaji3a8c86yfy5f4f9pzidmurvj82dz) using CNN model.

a. Include Embedding layer in the design of your models and report if that leads to a better performance

b. Plot loss of the model and report if you see any overfitting problem

c. What techniques you can apply to fix overfitting model

1. Implement text classification on the [20news\_group dataset](https://umkc.box.com/s/qhtpjz7hdb8vdzmwscp7alvs6ilt0zf2) using LSTM model.

a. Include Embedding layer in the design of your models and report if that leads to a better performance

b. Plot loss of the model and report if you see any overfitting problem

1. Implement image classification with CNN model, using one of the below datasets:

<https://www.kaggle.com/slothkong/10-monkey-species>

<https://www.kaggle.com/prasunroy/natural-images>

1. Report your classification result with and without doing scaling
2. Save the model and then predict on one of the test data. Report the prediction and check if it has been predicted correctly or not
3. The purpose of this question is to learn about text generation. Use [New York Times Comments and Headlines](https://umkc.box.com/s/v9d6l40pmqsu4x4h6edhvgxx8yn8h47d) to train a text generation language model which can be used to generate News Headlines
4. Pass a sample headline or word to the model and predict the generated headline
5. Apply Autoencoder on the **Cifar\_10** dataset and then pass the result of Autoencoder to CNN or LSTM or three layers model to classify data
6. Repeat the same thing with PCA (apply PCA on the dataset and then pass the result to CNN or LSTM or three layers model)
7. Compare the result of both approach

**Submission Guidelines:**

1. submission must be in a group of three or four students.

2. Submit your source code and documentation to GitHub and represent the work through wiki page properly (submit your screenshots as well. The screenshot should have both the code and the output)

3. Comment your code appropriately

4. Video Submission (2 –3 min video showing the demo of the project-exam, with a brief voice over on the code explanation)

5. Submit only report at Turnitin in UMKC

6. Remember that similarity score should be less than 15%

7. **Submit your work through the feedback form to the github**

7. The report should include below details

I. Introduction

II. Objectives

III. Approaches/Methods

IV. Workflow

V. Datasets (if applicable)

VI. Parameters

VII. Evaluation & Discussion

VIII. Conclusion

**Evaluation Criteria:**

1. Report similarly score (should be less than 15%)

2. Report Quality (check the below example reports for reference)

3. Time (should submit before due time)

4. Wiki page Example Reports:

<https://github.com/stratospark/food-101-keras>

<https://github.com/matterport/Mask_RCNN>

<http://blog.stratospark.com/deep-learning-applied-food-classification-deep-learning-keras.html>

**University Policy on Student Conduct:**

Cheating, plagiarism, disruptive behavior and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL:

<https://catalog.umkc.edu/special-notices/academic-honesty/>

<https://www.umsystem.edu/ums/rules/collected_rules/programs/ch200/200.010_standard_of_conduct>