=> Machine Learning Assignment 4 <=
100 MARKS

If you have any problems with this practical assignment, speak up well before the deadline!

Deadline

Submit all tasks on RuConnected by **the deadline**

Task 0: Upvote the Kaggle notebook [10]

Upvote the Kaggle notebook called IntroCNNsKeras, and I will check it against your username/email address.

Task 1: MNIST, but Fashionable! [25]

The Fashion MNIST dataset is a set of greyscale images of various types of clothing. Specifically, there are 10 different types of clothes.

Use this notebook as a starting point to classify items on the Fashion MNIST dataset using Keras.

Aim:

- Split the training data into train:validation split of **80:20**
- **Train** using training data but by evaluating the **validation** set
- **Maximize** accuracy on the **unseen** test set

Deliverables:

- Python Notebook with saved outputs

Task 2: Solve a real-world problem! [25]

Create a notebook that uses the Keras API to classify the Cover Type.

Aim:

- Split the training data into train:validation split of **80:20**
- **Train** using training data but by evaluating the **validation** set
- **Maximize** accuracy on the **unseen** test set

Deliverables:

- Python Notebook with saved outputs

Task 3: Transfer a model for better, faster training! [40]

Create a notebook that uses the Keras API to classify CIFAR10 images. Feel free to use IntroCNNsKeras or the updated version on Colab as a foundation to build a better model! *Hint: a smaller model may converge faster to a better accuracy.*

Aim:

- Split the training data into train:validation split of **80:20**
- You are limited to a maximum of **20 Epochs!**
- **Train** using training data but by evaluating the **validation** set
- **Initialise** a *better* **model** than ResNet50 for **transfer learning**.
- **Maximize** accuracy on the **unseen** test set
    - Use **transfer learning** with appropriate frozen layers for faster training.
    - Use **batch normalization** to improve training speed and accuracy.
    - Use **data augmentation** to increase dataset diversity.
- Produce (four) line graphs of training and test losses and accuracies on a single plot!

Deliverables:

- Python Notebook with saved outputs