# Mini-Project

Kaggle & Getting Started

## Kaggle.com:

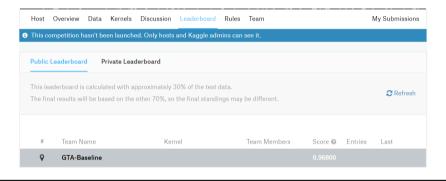
- Register on Kaggle
- Go to competition link
- Team Registration once signed up on Kaggle:

https://www.kaggle.com/t/3713b8edcaab4ac7ac6045d7353c1aba

# https://www.kaggle.com/c/acse-module-8-19 ACSE-Module-8-39 Legustooird for ACSE Modules of Competition 2019 Imperial College on tool This competition haan't been launched. Only hosts and Kaggle admins can see it. Overview ACSE-Module 8 Competition Track Introduction This is the Kaggle Inclass Competition for the Imperial College Applied Computational Sciences and Engineering Master's Course Mini-Project on Machine Learning. Dataset The dataset is a version of the Kuzushiji-MNIST The goal is to score a high accuracy on the prediction of the test set of images. Submissions You are allowed two submissions to the leaderboard per day. The shown leaderboard score is equivalent to 30% of a random subset of the test set. You are allowed to submit two final versions for scoring on the private leaderboard - this is what you will ultimately be ranked on. Pestrictions

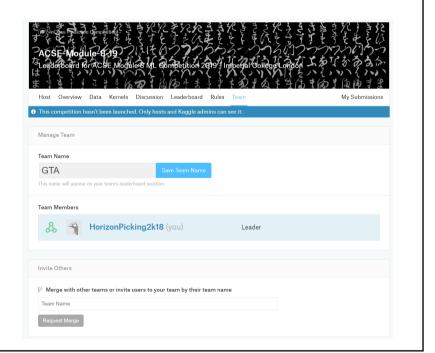
## · Leaderboard:

- Public: Accuracy evaluated on 30% of test set
- Private: Accuracy evaluated based on other 70% of test set
- Two Submissions per day per team
- Try to beat the GTA solution ©
- Leaderboard submissions reset at 1 am, but don't work till then
- Thursday, final two submissions selected for private leaderboard score



## Logistics:

- Make team
- At least 1 person/team
- 2 Submissions / day / team
- Use Submissions wisely
- Trust your Cross-validation
- Submissions reset 1 am
  - Don't work till 1 am
- On Thursday:
  - Select best two submissions
  - These two count!
    - -> Private Leaderboard



- Suggestions:
  - Make sure to tick boxes of outline
    - (Report, Reproducibility, Code Sustainability, ...)
  - Make sure the basics work:
    - Cross-validation
    - Submission format (.csv file)
    - Plot things where unsure or graph them
    - Build a working model, then expand
  - Things to try:
    - Simple Model + Add More Layers
    - Dropout
    - Batch Normalization
    - Data Augmentation

    - Early Stopping
      Other Optimizers e.g. RMSProp, Adam, SGD+Momentum
    - Transfer Learning
    - Better Network Architectures (as discussed in course): AlexNet-Style Models
  - You can use existing code, even from web, but expect to be questioned on everything!
  - Consider computational resources e.g. Colab only limited time for training until disconnect

