# Natural Language Processing
# COM3029 & COMM061

Coursework

## Table of Contents

## Topic Overview

Text classification is one of the most in-demand functions that has been implemented for solving a wide variety of problems. The most common scenario of text classification is for categorising documents in topics such as news items related to business, sports, politics or other events; another common scenario is for identifying spam in emails, or even understanding the sentiment on messages such as Twitter or when rating products on websites such as Amazon. During the prediction process (i.e. inference) the input text could eventually either be assigned with only one label (multi-class) or with multiple labels (multi-label), depending on how we need our application to behave.

Your task is to build a multi-label classifier prototype for a data topic of your choice. There should be at least 5 labels in the dataset and enough data (for each label) to train your chosen classifier model. The topic you choose does not affect the marking, so you should concentrate on ensuring that the dataset you select is appropriate for the task you work on. For example, as a topic, you might want to choose and classify:

- Genres of a movie description (e.g. comedy, drama, sport, horror, etc.)
- News article categories (e.g. business, sport, politics, technology, etc.)
- Inappropriate content in messages (e.g. bad language, violence, drugs, persuasion, and so on)

… or any other data topic you might find appropriate.

Some datasets you could consider (not exhaustive list):

- Fast.ai datasets (NLP specific)
- Kaggle datasets (e.g. Toxic Comment Classification Challenge)
- UCI dataset
- Mulan datasets
- Collection of datasets at Universidad De Cordoba (similar to Mulan)
- Many datasets also listed in Wikipedia

You can also build your own dataset by collecting text data from sources such as Wikipedia, News APIs or RSS feeds, Twitter, etc., but only do that if your team has the technical skills, time and resources to do so.

## Structure

The module is assessed 100% on this coursework. The coursework contains both group and individual contributions and is divided into three parts (including a separate submission for each):

1. A group declaration of work (i.e. a plan)
2. An experimentation part where each individual works independently, based on the group plan
3. A final part where a basic prototype system is put together by the group, based on the results of the individual contributions.

The group can be a minimum of 3 students and a maximum of 5. You will need to declare your group on SurreyLearn. It is recommended that you find your group as early as possible and talk about the coursework requirements. If you cannot find a group, you must tell the lecturer immediately so that you can be allocate to one.

**HINT 1**: The methodology is much more important than the accuracy of the model. So, make sure you appropriately follow the taught methods for performing the experiments rather than spending too much time improving the accuracy of the model.

**HINT 2**: Work continuously as a group, but make sure the individual part is works on and submitted separately. The team should be there to support any of the members that has issues, and in some cases the original plan can change if needed.

**HINT 3**: The deadlines are there for submitting that different required parts, but this should not stop you from moving forward to working in the next part! For example, if each individual has finished working on the experiments on week 8, then there is no need to wait till week 10 to build the prototype required for the final part.

**HINT 4**: Although I encourage you to search the Web for finding examples and researching solutions, and material such as code, data, definitions, diagrams, lecture slides, lecture labs, and other that will be used directly or indirectly in your submissions, must be referenced in a clear way.

## Group - Declaration

**Weight**: 10% of the marks
**Submission date**: Tuesday week 5

You will start this project as a group, then work individually and eventually get back together to complete the last part. This first part is important as it will set the group and work structure for the next two parts.

Discuss as a group and decide on the following:

1. Choose a relevant topic and explain why is relevant (2 marks)
2. Decide on an appropriate dataset and give a quick reasoning explanation (2 marks)
3. Plan the experiments, while keeping in mind your resources and time constrains, and explain the scope and priorities (2 marks)
4. Decide on a common development environment (code repository, python libraries needed, etc.) and list the choices made (2 marks)
5. Divide individual tasks and explain how you are planning to work together (2 marks). Each group member could potentially focus on a different experiment setup (i.e., this is part of the individual assessment). This might be experimenting with different:
    a. data set preparations (pre-processing and/or featurisation)
    b. algorithms
    c. pre-trained models (transfer learning)
    d. setup of the hyperparameters
    e. any other such relevant experimental variations (if needed)

### Deliverables

You will need to submit a one page document (min size 10 font) stating the group topic, the dataset, the plan for the experiments, the development environment and the individual tasks per group member. Any front cover or appendix can be added if relevant. This needs to be submitted by one of the group members on SurreyLearn, before the set deadline.

## Individual - Experimentation

**Weight**: 40% of the marks
**Submission date**: Monday week 10

Each student will individually research and experiment on different ways (or even the same way if the group feels is appropriate), to prepare data and train the model. The choice of the different individual experiments (tasks) should be discussed and decided as a group, as defined in the group declaration report, but attempted and documented separately by each individual. Group discussions and coordination should still continue. If the experimentation plan changes (needs to be a group decision), then you will need to just provide some justification. So, it is ok for the group to change the original plan without incurring any penalties. All experiments and documentation need to be done in a Jupyter notebook. Having multiple Jupiter notebooks for different experiments is also fine.

For the individual experiments, each student is expected to:

1. Analyse and visualise a dataset – produce charts and document observations (4 marks)
2. Experimentation with four different experiment setups, where you might be trying out different options such as (but no need to do them all):
   a. data pre-processing techniques – tokenise (e.g. will you use n-grams?), normalise text, apply stopwords, and so on
   b. NLP algorithms and techniques – explain choice
   c. text featurisation/transformation into numerical vectors – justify choices like one hot encoding, and other relevant methods
   d. training/text/validate dataset splitting – how did you split and why
   e. choices of loss functions and optimisers (if appropriate/relevant) – explain your choices with facts from the results
   f. other setups that you might find relevant – make sure to justify

   Since this will be subpart of a group's experiments – the implementation, methodology and critical thinking is what matters here, and not if you individually covered all possible experimental setups (20 marks, 5 marks per setup).
3. Train models (for each variation) – show details of experimentation for each experiment (4 marks)
4. Perform testing – show visuals such as confusion matrix or other relevant metrics for each experiment (4 marks)
5. Discuss best results and mention if there was any need to adjust and retrain during the experimentation (4 marks)
6. Evaluate the overall attempt and outcome – is the original problem solved? (4 marks)

Since some needed lectures won't be taught until late in the year, it is expected that you will still progressively and continuously work on the coursework. Each week there will be lab exercises that can help you with different parts of the coursework (e.g. data preparation, visualisation, data transformations, featurisation and other will be taught from week 2). So, it is not recommended to wait till all the lectures are taught before you get started.

### Deliverables

You will need to submit a Jupiter notebook documented appropriately, but this needs to be submitted in two formats: 1) as an ".ipynb" notebook file, and 2) as a pdf export in a way to show executed outputs. The notebook should contain visuals (where appropriate) to support tasks such as: label data distribution, histogram comparisons, text samples, classification accuracy curve, confusion matrix, etc. Additional notebooks or Python files can also be included, but make sure you zip the files together before submitting. If there are library dependencies, please also include a requirements file. This should be submitted by each student independently on SurreyLearn, before the deadline.

## Group - Deployment

**Weight**: 50% of the marks
**Submission date**: Wednesday week 12

This is the last part of the coursework assessment, where students get back together and combine their individual findings, choose and deploy the best solution, and build a pipeline that will train, deploy and monitor the model automatically.

All this work needs to be demonstrated and documented in a Jupyter notebook, together with any additional Python files that you might need to build. Having multiple Jupiter notebooks for different tasks is also fine.

Tasks:

1. Research different model serving option(s) and explain what would be the right choice for your case (5 marks)
2. Build a web service (10 marks) to host the chosen model as an endpoint (running that locally on your machine is sufficient)
3. Build some functionality to perform testing on the deployed endpoint and document findings in the notebook (10 marks)
4. Discuss the performance of the service you implemented, and justify the good and bad points (5 marks)
5. Build some basic monitoring capability to capture user inputs and the model predictions. There is no need to build functionality that will detect any concept drift. (5 marks)
6. Build a basic CI/CD pipeline that will build and deploy the model when data or code changes. There is no need to trigger this automatically, so a manual execution script will suffice. (5 marks)
7. 10 min screen recording (with voice description) of a demonstration of the group solution. The demonstration should show how you worked through the above tasks 1-6 (briefly) and it should clearly demonstrate the execution of the code (compulsory - 10 marks for the presentation clarity)

If for any reason none of the team members managed to complete the individual part of the assessment and you have no model to deploy, then you can either contact the lecturer to get a pre-built model, or alternatively you can use a model you found online. In either way, this should be documented appropriately in the notebook.

### Deliverable

You will need to submit a Jupiter notebook documented appropriately, but this needs to be submitted in two formats: 1) as an ".ipynb" notebook file, and 2) as a pdf export in a way to show executed outputs. Additional notebooks or Python files can be included, but make sure you zip the files together before submitting. If there are library dependencies, please also include a requirements file. The presentation recording file should be submitted in a popular format such as mp4, avi, mov, etc. This needs to be submitted by one of the group members on SurreyLearn, before the set deadline.