**Internship projects with LV=GI**

LV=General Insurance is a strategic partner for the University with a focus on Data Science. LV=GI has an opportunity for two three-month paid internships working with the Data Science Team based in their office in the Merchant Venturer’s Building in SCEEM. There is also the possibility of an additional unpaid internship if the demand is strong. The LV=GI Data Science team is 50 strong and 18 of them are based in the UoB office in SCEEM. They have a strong ethos of innovation and knowledge sharing and an inclusive and welcoming culture. A number of the team have joined via previous internship programmes and the team understands the challenges and opportunities of working on MSc level projects. You would have an academic supervisor in addition to day-to-day working with the LV=GI team.

**Hyper Parameter Tuning Frameworks on Parallelised Compute**

There are a variety of different approaches & frameworks for hyper parameter tuning (e.g. hyperopt, optuna etc.), some of these approaches are serial by nature but other methods could be parallelised (e.g. across a Spark cluster or similar). We would like to explore which approaches work best for different model types & where we could best take advantage of parallelised methods. This should include an analysis of total training time and any other considerations related to implementation (e.g. cost, compatibility).

**Productionising ETL pipelines:**

As the software engineering field progresses, it has established a wealth of best practice around code development, documentation and testing. Teams writing data transformation logic have been slow to adopt these best practices although there is reason to believe that they can bring just as much value to this domain. Some tools have recently been developed to enable software engineering best practice in the data domain (e.g dbt). We would like to research the best way to apply software development best practice to ETL pipelines while answering questions such as:

1. How can we sufficiently test code and datasets?
2. How can we document the business logic behind the transformations?
3. How can we version track the data produced?
4. What tools should we be using?

**Automated Machine learning**

A comparison of AutoML, auto hyper-parameter tuning & auto feature engineering Python frameworks (e.g. ATM, FeatureTools etc) to establish what works well in which circumstances. This should include an analysis of training times, model choices, feature choices, any considerations for implementation (i.e. compatibility, model explainability) & when it might be appropriate to use one these packages for a particular use case due to a particular strength/weakness in a certain area.

**Model inference optimisation**:

Pandas has been the go to package for data transformation in python for a while now, although recently packages like polars ([GitHub - pola-rs/polars: Fast multi-threaded, hybrid-out-of-core DataFrame library in Rust | Python | Node.js](https://github.com/pola-rs/polars)) have emerged touting better performance in large data scenarios like: [Database-like ops benchmark (h2oai.github.io)](https://h2oai.github.io/db-benchmark/).

We would like to see If similar benefits can be realised within a model inference setting, where instead of large data there is typically only a single row.   
We would like to understand the benefits and trade-offs between different data transformation package and the potential for this to increase throughput of current hardware and the potential cost reduction.