The aim of my first project was to create a large dataset of technology companies. I was given a set of xml documents containing search results from a web API. Each document represented a search for a particular company, but contained around twenty results, so the first task was to extract the result corresponding to the desired company. I did this by generating a set of possible subdomain names given the company name (for example, ‘Electronics Incorporated’ might have subdomain electronicsincorporated, electronics, ei, electronicsinc, etc), scoring the URL’s from that company’s xml doc by their similarity to these possible subdomains, and picking the URL with the highest score. The winning URL’s homepage was pulled, along with its ‘about’ and ‘contact’ pages, if they existed. This left us with text from three thousand or so websites.

The task was then to classify the pages as belonging to a particular type of tech company or not, using machine learning methods. My idea was to find a subset \( S \) of a given kind of tech company using clustering, and then to apply the methods of semi-supervised learning found in [1] to the whole dataset, using \( S \) as the set of positive examples. The clustering was done by the Ward hierarchical method, using the rows of the pages’ TF-IDF matrix as the data points. To find a cluster’s defining features, I treated membership and non-membership of the cluster as two class labels, and found the terms with the highest information gain under this labelling. With this method, extracting pages likely belonging to a given type of tech company was possible via a two-step process: 1) find pages that belong to organisations related to the desired category by selecting clusters whose defining features are terms related to that category, and 2) from these, find pages that belong to technology companies by selecting clusters whose defining terms are ‘technology’, ‘data’, ‘software’, etc. With a set of positive examples in hand, further tech companies of the required type could be found by training a support vector machine iteratively as in [1].

I also extracted addresses for as many of the companies as possible. This was done with the help of a data set containing a large number of settlements and their countries [2]. The basic idea was to check each word in each document against the list of countries, and check whether the word (or pair of words) occurring directly before any match was the name of a settlement in the relevant county.

My second task was to produce synthetic medical records for cancer patients, as a resource for testing medical software and models. The records were generated according to FHIR standards [3]. They reproduce real macroscopic incidence statistics: the relative incidence and age distribution of cancers in the records matches that in the UK population. The simulated breast cancer patients all come with a FHIR questionnaire detailing their family history. This is used to calculate the event density \( f(t) \) for contracting breast cancer according to the two-loci genetic model proposed in [4], and the patient’s age at diagnosis is effectively sampled from the distribution \( f(t) \). Lifestyle risk factors (e.g. smoking, alcohol, diet), again stored in FHIR questionnaire format, were also taken into account when building the distributions from which the diagnoses were sampled; the data used for this feature can be found in [5]. I also generated diagnostic test results, again in such a way as to reproduce high-level statistics on staging at diagnosis. These statistics were taken from [6].
From the first project I gained competence in/ an understanding of:

- Python’s syntax and philosophy
- Web scraping methods and etiquette
- The BeautifulSoup library, for parsing and navigating html/xml
- Document vectorization/ TF-IDF
- Scikit-learn, Python’s machine learning library
- Hierarchical cluster analysis
- Information gain
- The Random Forest algorithm
- The Support Vector Machine algorithm
- Machine learning on positive and unlabelled data
- Some fundamental ideas in statistical learning

From the second project:

- Medical interoperability frameworks
- Survival analysis
- Computational genetics
- Pandas, a package for manipulating data
- Clinical coding, eg SNOMED, ICD 10
- Object oriented programming
- Converting between JSON and xml formats

Other general skills gained:

- Familiarity with various data structures, eg JSON
- IPython terminal
- Readable programming: PEP 8 style
- Statistics
- Regular expressions

References: