Reflection 5

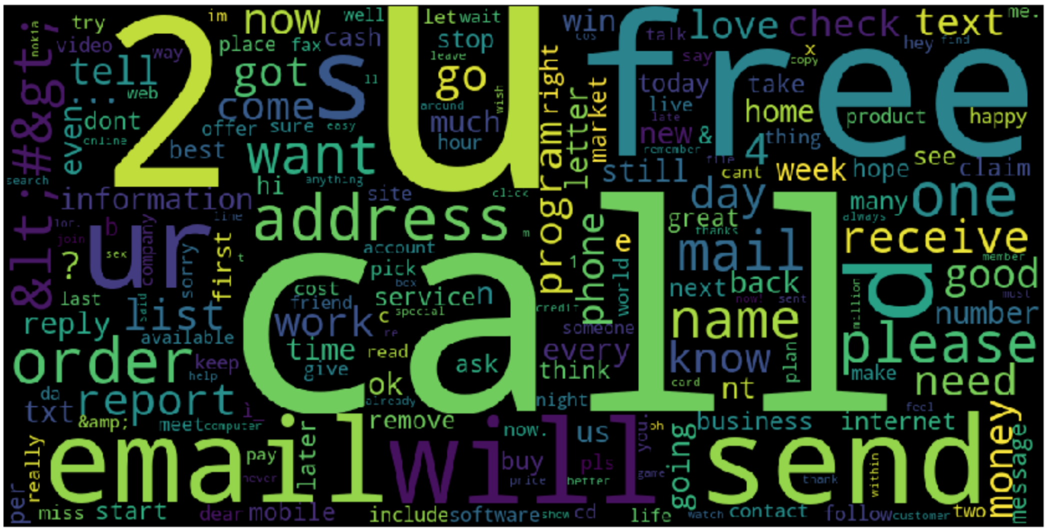
Sydney Vertigan

June 21, 2019

For this project we wanted to try something we hadn't done before. Our aim was to encrypt and decrpyt messages using parralelisation.

We all found our datasets and worked out how to use pyspark together. That made us realise what we may and may not be able to do with our messages. Myself and Sam both went away over the Easter break so Junfan started the main project and getting wordcounts and relations between words in the texts. Once we were back we all worked on it together on one laptop and we found it hard to set chunks each as we felt everything should be done in order for this project as we weren't always sure where we were going with it. Below shows the order we did things in, why and any problems we faced.

We found a datasets consisting of emails and books. First we wanted to be able to detect between fraudulent messages and normal ones. We could see the different types of emails and compare them to the 'normal text' of the books. It was interesting to see that spam emails' most frequent words were those you would expect, illustrated in the Wordcloud below.



To be able to detect fraudulent messages we tried a few different machine learning algorithms. Random Forests seemed to work well for parallelisation. We think this is because each tree is being trained independently. However, we could not get this to work well for the whole dataset. Many times our memory was not large enough to process it in this way.

Our aim is to be able to encrypt and decrypt messages using pyspark. We chose to use Fernet to do this. This is because it seemed the most widely applicable symmetric encryption library and can even support key rotation implementation. To make the process fast we used a vectorised form of the data and this worked well for speed. We also wanted to see if this way of doing things would work for unbalanced data - which it did handle very well.

We did have a lot of problems using pyspark and not being able to do things which we would find ‘easy’ using numpy or pandas. Also the setup of pyspark on everyone’s computers took sometime. One thing that was very annoying is that if you get one bad datapoint then that will change the whole processing.

Overall, we found that parallelisation is not always the answer - it just doesn’t work for some processes. However, some things are much faster. One thing that stands out is that Junfan implemented the Jaccard distance matrix (something he and myself had done on a previous project together), using parralel processing and vectorised computation and it was much faster! 17 times faster to be exact. However, the big flaw was that reading big data from the hard disk could be very slow.

I’m sure if the project wasn’t over exam time we would have delved deeper and came to some better conclusions. Moreover, I do feel we have learnt a lot about what can and cannot be done in pyspark and also, problems we could face when using it in the future.