**Report**

The purpose of this project was to to write a generator for data augmentation of transaction data. The augmentation I used covered two features namely: the ‘date’ and the ‘amount’.

While dealing with the problem I always kept in mind that we want to perform small random transformations to the data that are similar to those we see in real life. Augmentation techniques are very popular in computer vision but for numerical data is not that developed. Since I have never used augmentation techniques for numerical data before I had to use common sense in order to decide how I will augment the features.

In this occasion, the augmentation of the ‘amount’ feature of the transactions was very straightforward. I randomly increased or decreased the value of the original ‘amount’ by a range of +-40% of the original value. This would be enough to alter the numerical value and keep the context of the transaction.

The augmentation of the ‘date’ feature was a bit trickier. I had to transform the string to a datetime object and then to a timestamp object in order to alter it numerical. Then after the value was altered, I transformed it back to a datetime object and then back to timestamp and eventually string.

Finally, I found an interesting way to introduce the probability of augmentation or not for each augmentation technique. I wrote a function to return either ‘A’ for augmented or ‘NA’ for non-augmented just like a flip of a coin. But here the coin can be biased, and this is where the probability p is coming into play. This function will be called for each augmentation technique to assure we are using independent probabilities.

**The ‘bud.py’ file contains extended commentary for every line explaining the purpose and the functionality of the code.**

**A ‘demo.py’ script has been included as a demonstration of the Generator class. Run this script to automatically generate augmented data.**

I enjoyed this challenge immensely and I look forward to discussing it with you.