Zastanawiałem się nad wyborem pomiędzy:

* Stworzeniem inicjalnej koncepcji/postaci DAG’a w Airflow w ramach zadań, które wykonywałem jako członek zespołu projektowego wdrażającego tzw. *data lake* dla banku Pekao S. A. Kolejne nowe DAGi wzorowały się na mojej inicjalnej koncepcji.
* Wygenerowaniem predykcji szeregów czasowych poboru energii dla PKP Energetyka.
* Stworzeniem binarnego klasyfikatora dla jednej z firm ubezpieczeniowych w oparciu o tzw. zdarzenia rzadkie.

Ze względu na jednoczesne wystąpienie obu wymiarów wyzwań pracy z danymi, z którymi musi mierzyć się tzw. *Data Scientist*, postanowiłem opisać punkt drugi – predykcja szeregów czasowych.

Osiągnięcie celu wymagało przetworzenia danych do postaci, która była odpowiednia do zastosowania bibliotek Python’owych, używanych do budowy modeli predykcyjnych.  
Otrzymane dane charakteryzowały się postacią tabelaryczną zbliżoną do tzw. tabel przestawnych w Excel. Były to tzw. tabele *wide* a nie *long*. Dane dotyczące pobory energii zbierane były w interwale 5 minutowym. Było ich dużo.

Ze względu na konieczność transponowania danych do postaci *long*, dla dalszych czynności związanych z tzw. preprocessingiem (np. agregacji danych do interwałów 15-minutowych lub 60-minutowych), wymagany był *batch processing,* oparty o partycje. Oparłem się o partycje wyrażające datę i czas.

Bez podejścia iteracyjnego, opierającego się o partycje przetwarzanie danych napotykało na problem związany z brakiem wolnej pamięci, tzw. *out of memory*.

Po udanej agregacji należało dokonać mądrego wyboru algorytmu dokonującego predykcji. Celem zadania była predykcja poboru energii dla X regionów i Y lokacji na cały rok „w przód” w oparciu o dane historyczne z dwóch poprzednich lat.

O ile dobrze pamiętam zdecydowałem się zastosować głębokie sieci neuronowe LSTM (LSTM – *long short term mermory*). Mechanizm predykcji opierał się na ruchomym *input*’cie o oknie 7 dni, który prognozował kolejne 7 dni. Mechanizm przesuwał się o jeden dzień i prognozował ponownie. Eksperymenty przeprowadzałem wspierając się biblioteką Keras.