DS-670 Lab 9: Apply Aggregation and Group operations to my dataset.

Mohamed Mohamar

1. complete reference to the IEEE or ACM paper you are trying to outperform. Only articles published by IEEE or ACM will be accepted.

F. Lollia, R. Gamberinia, A. Regattierib, E. Balugania, T. Gatosb, S. Guccib. International Journal of Production Economics. Volume 183, Part A, January 2017, Pages 116–128

2. The results of the paper I have chosen.

"Managing intermittent demand is a vital task in several industrial contexts, and good forecasting ability is a fundamental prerequisite for an efficient inventory control system in stochastic environments. In recent years, research has been conducted on single-hidden layer feedforward neural networks, with promising results. In particular, back-propagation has been adopted as a gradient descent-based algorithm for training networks. However, when managing a large number of items, it is not feasible to optimize networks at item level, due to the effort required for tuning the parameters during the training stage. A simpler and faster learning algorithm, called the extreme learning machine, has been therefore proposed in the literature to address this issue, but it has never been tried for forecasting intermittent demand. On the one hand, an extensive comparison of singlehidden layer networks trained by back-propagation is required to improve our understanding of them as predictors of intermittent demand. On the other hand, it is also worth testing extreme learning machines in this context, because of their lower computational complexity and good generalisation ability. In this paper, neural networks trained by back-propagation and extreme learning machines are compared with benchmark neural networks, as well as standard forecasting methods for intermittent demand on real-time series, by combining different input patterns and architectures. A statistical analysis is then conducted to validate the best performance through different aggregation levels. Finally,

some insights for practitioners are presented to improve the potential of neural networks for implementation in real environments."

3. Describe your results so far, and your next objective in terms of data analysis.

   So far at this point, I am calculating the weights for the neural network, applied to my stock market data, to predict the market volatility. 75% of my dataset is used for training, 15% for testing. I will then apply the model to at least three to four different sectors.

4. Describe how you will use the concepts of aggregation and group operations

   My results generated in Zeppelin are uploaded in the attached PDF file.

5. create a table showing the code for every operation in the left column and the time measurements for every operation in the right column. You need at least 10 operations. Highlight in bold the lines of code where you use aggregation or group operation concepts.

Codes	Time (in Seconds)
%pyspark df = DataFrame({'category': ['a','a','a','a','b','b','b','b'], 'data': np.random.randn(8), 'weights': np.random.rand(8)}) df	0.00297403335571
grouped = df.groupby('category') get_wavg = lambda g: np.average(g['data'], weights=g['weights']) grouped.apply(get_wavg)	-0.00128388404846
close_px[-4:]	0.0018208026886
rets = close_px.pct_change().dropna() spx_corr = lambda x: x.corrwith(x['Close']) by_year = rets.groupby(lambda x: x.year)	0.00271677970886

by_year.apply(spx_corr)	
by_year.apply(lambda g: g['High'].corr(g['Low']))	-0.00465798377991
by_year.apply(regression,'Close',['Volume'])	-0.00968909263611