Overview and Instructions

# Description

The IPython notebooks and Python file in the attached address the four prompts in the Data Science Challenge (DSC). Each notebook contains relevant scripts and thorough documentation of my results, approaches, assumptions, questions, and next steps. The notebooks begin with the DSC prompt and package imports. These preliminaries are followed by a summary of my conclusions for each question. Running the subsequent cells in each notebook will re-produce the steps that I took to generate those conclusions.

# Summary of results

The four notebooks demonstrate my ability to load, visualize, and wrangle data, as well as my approach to predictive modeling. The predictive model I developed was able to reduce the estimated cost incurred by fraudulent transactions by 29% relative to the baseline case of having no fraud detection. I approximated the cost of fraud using utility function based on a literature review of Capital One fraud prevention efforts. The utility is function is a simple approximation of a complex business case, but indicates that the framework I developed could provide business value in this synthetic setting. Although I was limited to relatively simple models by computational resources, my approach to model evaluation and feature development was sufficient to create a measurably useful model.

# Submission files

Every notebook has a corresponding HTML file that visualizes my documentation, scripts, and results. The file, data\_load\_clean.py, contains utility functions I created based on the answers to Question 1. These functions are called in the notebooks for Questions 2-4 and assume that the data\_lead\_clean.py file is in the same directory as the notebook. Because the DSC asks for the submission to not include the original data, the scripts assume that the data is stored in the parent directory with the name transactions.txt. Finally, the Feature Evaluation directory contains the visualizations and statistical analysis used in feature generation and selection. These are stored outside of the notebook for Question 4 for ease of use.

# Packages used

I leveraged common data science and scientific computing Python packages. These packages are cited below:

* McKinney, Wes. "Pandas, python data analysis library." URL http://pandas. pydata. org (2015).
* Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
* Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272.
* Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021
* J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
* Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
* Taskesen, E. (2020). distfit - Probability density fitting (Version 1.4.0) [Computer software]. https://erdogant.github.io/distfit
* Fernando Pérez, Brian E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: https://ipython.org