Dylan Labatt Randle

dylanrandle.github.io

Email : dylanrandle@g.harvard.edu
Mobile : +1-647-641-1994

Education

- **Harvard University, School of Engineering & Applied Sciences** — Cambridge, MA
  Master of Science in Data Science; GPA: 4.0 — *Sep 2018 – Present*
  - **Distinctions**: Scholarship in Applied Computation
- **University of California at Berkeley, College of Engineering** — Berkeley, CA
  Bachelor of Science in Industrial Engineering & Operations Research; GPA: 3.9 — *Sep 2012 – May 2016*
  - **Distinctions**: High Honors, Phi Beta Kappa, Tau Beta Pi

Experience

- **Harvard University** — Cambridge, MA
  Graduate Researcher — *Nov 2018 – Present*
  - **Deep Learning for Turbulence**: Researched and developed methods for solving Reynolds-Averaged Navier-Stokes (RANS) equations with neural networks in a fully unsupervised manner. Tech stack: Python, PyTorch, Harvard Odyssey Supercomputer
  - **GANs for Differential Equations**: Researching methods for training Generative Adversarial Networks (GANs) and ensembles of neural networks to solve differential equations without access to ground truth data. Tech stack: Python, PyTorch, Google Colab
- **Amazon Robotics** — Boston, MA
  Data Science Intern — *Jun 2019 - Aug 2019*
  - **Data Engineering**: Built automated, scalable data pipeline for big data queries (500+ billion rows), storage, cleaning, and transformation. Tech stack: Python, Apache Spark/Hive/Hadoop, AWS EMR/S3/Athena
  - **Data Science**: Developed flexible & modular machine learning library for proprietary internal project. Built API for data filtering, feature selection, training, tuning, and testing models (e.g. linear models, gradient-boosted trees, feedforward neural networks). Developed visualizations and interpretability algorithms (e.g. accumulated local effects, Shapley additive explanations) for model explanations. Tech stack: Python, AWS SageMaker, JupyterLab, Pandas, NumPy, scikit-learn, XGBoost, Keras
  - **Data Product**: Prototyped interactive user-interface and developed example use cases allowing technical and non-technical users to easily train machine learning models for a broad set of tasks. Reduced analysis time from weeks to hours. Presented project to teams from across the organization. Tech stack: Jupyter Notebook/Widgets
- **Hubdoc** — Toronto, Canada
  Lead Data Scientist — *Jan 2017 – Jul 2018*
  - **Production Deep Learning**: Developed and deployed production deep learning system using LSTMs & CNNs for entity extraction and text classification of financial documents. Models trained on over 10 TB of text and image documents. Reduced extraction time from 24 hours to 5 seconds with highly scalable, asynchronous pipeline. Cost savings estimated at $2MM/year. Tech stack: Python, Keras, Tensorflow-Serving, AWS EC2/S3/KMS, PostgresSQL, RabbitMQ
  - **Data Science**: Conducted business and engineering analyses: e.g. prediction of labor requirements and anomaly detection of web scrapers. Wrote reports and built data visualizations for company intranet. Tech stack: Python, NumPy, scikit-learn, JavaScript, D3.js
  - **Leadership**: Regularly presented results and recommendations to C-suite. Integral in crafting team strategy and roadmap. Involved in fundraising and presentations to investors. Delivered machine learning lecture to 60+ people

Projects & Research

- **Deep Learning for Differential Equations**: Research work completed as member of Protopapas research group on deep learning for solving differential equations
- **Tensorflow with Spark**: Training recurrent neural networks on a 1.5 TB dataset with Tensorflow on a Spark/Hadoop cluster with AWS Elastic Map Reduce
- **Modeling Microbiome Dynamics**: Modeling Granger causality with causal-LSTM model of high-dimensional experimental microbiome time-series data from mice

- **Automatic Differentiation Package**: Python package implementing automatic differentiation, supporting both forward and reverse modes; stochastic gradient descent and Adam optimizers implemented as example use cases
- **Twitter Troll Classification**: Project achieving 96% accuracy classifying Twitter trolls using tweets scraped from accounts indicted for meddling in the 2016 U.S. elections

Programming Skills

Languages: Python (NumPy, Pandas, matplotlib, scikit-learn, Keras, PyTorch, PyMC3), SQL, C
Technologies: AWS (EC2, EMR, S3, Athena), Apache Spark/Hadoop, Git, Jupyter, OpenMP/ACC