**James Littiebrant**

Python Programmer Analyst

Indianapolis, IN - (608) 334-6902

jameslittiebrant@gmail.com

**Summary:**

* An applied data analyst and python developer with extensive knowledge of not only machine learning techniques, but also statistical/probabilistic approaches to hypothesis testing, mathematical foundations for neural networks and APIs like Keras which are built on packages like Theano and Tensorflow, and low level programmatic optimization and vectorization techniques.
* Believe in actionable results driven by good initial questions about the data with concrete evidentiary backing that not only guides decision making, but springboards new ideas. This means, that data is not fed into linear algebra machines and results are obtained.
* For data science to be conducted, scientific testing principles must be used to evaluate all results. To achieve results, not just machine learning results, but hypothesis testing, requires programmatic skills, and data science tools, mathematical understanding of how the models work, and a thorough grounding in how hypotheses relate to the data at mathematical and real-world levels.

**Education:**

**University of Wisconsin – Madison, 2008-2013**

* Degrees in both Philosophy of Science and Mathematics
* Mathematical focus on group theory, and number theory

**Microsoft Certification in Python Programming for Data Science (From edX)**

* Six-week course in fundamentals python machine learning toolkits
* Focused primarily on Numpy and Scikit-Learn

**Achievements:**

* Thesis under Elliot Sober (recently ranked 1st in the world for philosophy of science) concerning the extension of bayesian hypothesis support updating based on empirical data.
* The caveat is that instead of individual hypotheses with known priors, this related to collections of hypotheses without the ability to determine any priors for the set.
* Numerous graduate classes in philosophy of science and advanced logic ranging from first order logic to Kripke semantics in modal logic

**Technical Skills:**

* ***Languages:*** R (2 years), Python 3.3-3.7 (4 years), Julia (1 year), Java 8 (6 months), C (indeterminate), (some Scala - 3-6 months)
* ***Database:*** T-SQL, MySQL, Postgres - queries, sprocs, SQL Server 2012/2016, Postgres 10, MySQL, Normalized data representation in relational databases
* ***Dataframe tools:*** Dplyr, Pandas, Numpy, SciPy sparse and dense matrices
* ***Vendor Tools:***SQL Server Management Studio, Pycharm, Tableau Public, Microsoft Office Suite, Jupyter Notebooks, Git, Linux (user since 1999 and Debian derivatives since 2007), Windows XP - 10
* ***Programmatic Tools:*** Pandas, Numpy, Scikit-Learn, SciPy, Dplyr, StatsModels, Keras, Theano, Tensorflow, PyMC3, XGBoost, LightGBM, SQLAlchemy, pyODBC connections, Matplotlib, Seaborn, Bokeh, ggplot2, NLTK, Orange
* ***Data Analysis Knowledge:*** Scikit-Learn suite (Certification from Microsoft 2017), Large and Small dataset statistical analysis and hypothesis testing, A/B testing techniques, randomization of trials, advanced feature engineering, end-to-end data visualization for exploratory data analysis to final report, pipeline automation for continued in-production use, vectorized optimization techniques, linear algebra transforms and applications, Numpy-Pandas mathematical interfaces, web scraping, data collection and merging from multiple sources
* ***Project Data Sources:*** Web scraped text (unstructured), economic housing data (structured, but dirty), employee behavioral data (structured, but limited in scope), campaign finance contributions (unstructured, dirty, incomplete)

**Professional Experience:**

**Cortland Capital Market Services (Part of the Alter Domus Group) – Indianapolis, IN**

**Oct 2017 – Present**

**Python Programmer Analyst / Data science**

**Description:**

* Cortland is a back office financial services company. It is often used by small funds to manage their Backoffice requirements and larger firms who require third party audit structures. The clients range from real estate funds to international hedge funds. Reconciliation technology is focused on developing automation systems and extraction techniques to reconcile transactions recorded at banks and clients.

**Responsibilities:**

**Behavioral analysis of business unit reconciliation behavior for more robust and representative reporting metrics (ReconStats)**:

* Took the initiative to develop an advanced mathematical model (3rd order calculations) to go beyond what the means and medians said to identify and quantify day to day behavior of manual reconciliations done by business units.
* Called the analysis the “communal guacamole bowl” model.
* Massive feature analysis and extrapolation from just two offset time series
* Exploratory graphing and analysis to refine and define what the data is capable of revealing and what are its limitations.
* Ground-up design of mathematical formula to calculate a variety of statistical measures to be selected from by managers for further business-unit side reporting.
* Relies on multiple server connections to extract the data, apply the behavioral analysis, and then write it efficiently to a snapshot database for verification and documentation purposes.

**Designed and deployed the SQL Server ETL script (PyMAM):**

* To enhance existing functionality of vendor software to improve speed and management of operations workflows. This was originally an SSIS project, but the demands for in-memory calculations proved too demanding and cumbersome to maintain for any period of time.
* Python-SQL Server interface to extract the data and after transformations and logic application, apply updates and inserts to automatically matches in reconciliation.
* Required deep analysis of existing stored procedures with nearly no documentation surrounding it, and often required reverse engineering both vendor store procedures and additional custom / ad-hoc stored procedures implemented by the team to account for incompatibility between source data and the software’s logic.

**Eaton Vance 1 & 2 onboarding project:**

* Designed, built, support, and end-to-end deployed the back end for the largest deal in company history. It was a sophisticated system of python scripts and postgres to properly identify required files for processing (total of 80 over the project) and then connecting those scripts through the restful interface of our execution system.
* Installed months earlier (Rundeck). Then the execution phase creates 4 normalized files for easy ingest into our reconciliation auto-matching software. Due to the extreme variability and unpredictability in source data, including critical missing data, multiple sql-python systems had to be developed to create accurate day-over-day reconciliation.
* Python-Postgres interface with snapshotting data for historical tracking, debugging, and day-over-day differencing calculations to extract required trade data.
* Multiple independent scripts organizing, collating, and executing the entire reconciliation ingest requiring both polling execution and an event-driven pipeline to flow the information through.

**Created YAML scriptable parsers for bank data extraction:**

* Created YAML scriptable so that single scripts could use the same code for each file but be dynamically configured at runtime to extract required data. This significantly improved speed of development and deployment, abstracted configurations from underlying code so that non-python developers could create extractions, and safe storage of the configurations in a SQL table for a single point of access.
* Required advanced object-oriented principles to create dynamically configurable parsers for a variety of data formats
* For different file structures, different OO design principles had to be selected and developed - don’t force data into code, code to data.
* Created automated data comparison tools and specialized testing environments to enable TDD implementation in the workflow. One of the data comparison tools for Pandas dataframes allows for rapid comparison of old and new data after code changes to identify any problematic data changes and side by side comparison of cell data in a sparse format for easier human readability.
* Required clear, fast, vectorized code to create outputs in a timely fashion
* Bank data, while formatted, could very easily be described as unformatted data considering the impoverishment and mutability of much of the data. Tracking changes to code and impacts on outputs with dynamically typed languages like Python/Pandas is cumbersome when several million cells might be involved
* This involves tracking code changes over potentially 100 different deals and more in the future so QA/QC has to be automated as much as possible
* Significantly reduced debugging times and reduced the creation of bugs over more manual processes by up to 10 hours a week.

**Environment:** Postgres, Windows Server 2012/2016, SQL Server 2012, Python 2.7, 3.5, 3.6, Windows 10, Pycharm

**Educational Communications Board (Engineering side of Wisconsin Public Television) – Madison, WI**

**Jun 2014 – Jan 016**

**Transmission Operator**

**Wisconsin Public Radio** – Madison, WI

**Jan 2014 – Jan 2016**

**Radio Operations Controller**

**Description:**

* Educational Communications Board (ECB) is the state-owned portion of public broadcast within the state of Wisconsin and is responsible for television and radio broadcasts, as well as NOAA alerts and the Amber Alert system. Wisconsin Public Radio is the University of Wisconsin owned branch and is the content source, but the two are essentially the same entity as affiliates of PBS and NPR. While working for them, my duties ranged from

**Responsibilities:**

* Transmission feed monitoring, data collection and archival database upgrades, occasional electrical engineering duties associated with transmitters
* Python related development with Raspberry Pis to enhance and develop our fiber network’s security. was primarily focused on WireShark network data collection, analysis, and low-level python network development to troubleshoot and improve existing systems.
* Management style was horizontal, practiced radical transparency, and the skills
* Acquired from my managers are more important to me than my development as a programmer.

**The Project Lodge – Madison, WI**

**2008 – 2011**

**Chief Audio Engineer**

**Description:**

* The Project Lodge (also known as Prolo) was a music venue on E Johnson St in Madison, WI. became their chief audio engineer after working with them for only 4 months and displayed an aptitude for sound despite not having prior experience.

**Responsibilities:**

* Developed a thorough understanding of signal processing and a firm argument for why CDs are better than LPs.
* Managed and training up to 10 other engineers at a time.
* Expanded our capabilities for live recording and became a mix/master engineer too.
* Ran the sound board for live events, 2-5 times a week.
* Maintaining the PA system, researching upgrades and implementing new tech.
* DIY get-it-done for any problem without an available solution.
* Everything else that it takes to run a small business including fixing the plumbing and mopping up.

***Personal Projects and Volunteer Work***

**OBI Studios – San Jose, CA**

**2016**

**Volunteered Work**

* Volunteered to work with Dave, who started the large band-stage practice and recording studio
* Trained him on software, cleanup and processing techniques, room balancing
* Did a mastering of the San Jose Community Band’s 2016 concert
* Did occasional live events when it was rented out and an experienced hand was needed

**Mutation Testing Development – Currently designing a system of decorators to dynamically alter data inside the ETL process.**

* It provides stress testing of the ETL process by varying data types, data structure, and table structure.
* Will allow automated, reproducible, and proactive bug detection in pipeline code when working with highly variable data
* Also have prototypes to save out processes which capture faulty or problematic data for QA in real time.
* My goal is to remove the need to modify existing code and move to YAML scriptability for testing procedures.