**Jvalin Dave**

**682.215.5246 |** [**jvalin17@gmail.com**](mailto:jvalin17@gmail.com) **| Philadelphia, PA**

|  |  |  |
| --- | --- | --- |
| **LinkedIn:** [linkedin.com/in/jvalindave](https://www.linkedin.com/in/jvalindave) | **GitHub:** [github.com/jvalin17](https://github.com/jvalin17/) | **AngelList:** [angel.co/jvalin-dave](http://angel.co/jvalin-dave) |

**TECHNICAL SKILLS**

* **Proficient:** Python, Flask, Pandas, Scikit-learn, NumPy, MLib, C, SQL, MySQL, AWS (EC2, S3, Lambda, SWF), Git
* **Exposure:** PySpark, Hadoop, Java, Ruby on Rails, D3.js, C++, Jenkins, Heroku, PostgreSQL, JavaScript

**WORK EXPERIENCE**

**HeavyWater Inc.** **| Machine Learning Engineer | Philadelphia, PA** Apr. 2017 - Nov. 2017

* Optimized datasets for machine learning algorithms by creating stratified 10-fold datasets in Python, achieving accuracy of 69% in classification of documents.
* Reduced execution time of optical character recognition (OCR) algorithm by over 50%, when replacing Java CLI with Java API, improving throughput by 2% and while reducing infrastructure cost by 7.5%.
* Implemented date extraction algorithm using Java, resulting in improving accuracy of date extraction by 17%.
* Automated AWS service infrastructure using Java, Python, EC2, Lambda, and S3, resulting in 25% reduction in processing time of insurance documents.
* Increased production throughput by 100% in 2 months by debugging 7 workflows using Linux commands and AWS Lambda to transfer data from EC2 instances to S3 buckets in order to avoid timeout failures.
* Led team of 3 engineers, minimizing overall cost of production by 20% and managed DevOps team based on agile development using CI/CD tools to maintain stable release of product on schedule.

**Lumidatum Inc.** **| Machine Learning Engineer | Dallas, TX** Nov. 2016 - Feb. 2017

* Designed self-learning recommendation engine using Python, Pyspark, MariaDB and AWS S3 resulting to 5% increase in sales through providing better targeted products to consumers.
* Engineered data pipeline through daily extraction of 30,000 lines of text from log files utilizing Python and MariaDB, automating system to optimize preprocessing time by over 67%.
* Analyzed and designed new machine learning algorithm to calculate lifetime value (LTV) of a customer using Python, D3.js and Matplotlib, resulting in 40% more accurate prediction.

**EDUCATION**

**Master of Science in Computer Science,** *University of Texas at Arlington* May 2016

**Bachelor of Engineering in Computer Engineering,** *University of Pune* Aug. 2013

**SIDE PROJECTS**

**Healthometica | Software Architect |** [**live**](https://youtu.be/ao3MGrbfs8E) May 2016

*Self-monitoring health system for University of Texas students to track effective remedies for treating daily issues.*

* Architected platform using UML diagrams and analyzing technologies such as Java, JavaScript, MySQL, JDBC, and Lucidchart for prototyping to decide on most efficient tools.
* Designed schema for MySQL database and JavaScript UI with sketching and surveying 40+ university students.

**E-mail Spam Detection | Machine Learning Engineer |** [**code**](https://github.com/jvalin17/Machine-Learning-algorithms/tree/master/E-mail%20Spam%20Detection) Dec. 2015

*Machine learning algorithm trained to analyze inbox emails to determine whether they are important or spam messages.*

* Engineered e-mail spam detection algorithm using Python and MySQL, resulting in 82.6% average accuracy.
* Built training and testing data using random function to select rows to ensure split is unbiased using Python.
* Determined most efficient algorithm and built Naive Bayes Classifier on training data of 4,000+ emails from the ground up without the use of machine learning algorithms in Python.

**Yelp Dataset Challenge | Machine Learning Engineer |** [**live**](http://dtyelp.github.io/Yelp-Data-Set-Challenge/) **|** [**code**](http://dtyelp.github.io/Yelp-Data-Set-Challenge/) May 2015

*Predictive natural language processing (NLP) in text mining for locating most commonly used terms in Yelp reviews.*

* Gathered over 1.8 million reviews utilizing Yelp API and chunking mechanism, refining reviews based on industry and place, and generating training and testing data for machine learning algorithm.
* Applied Naive Bayes Classifier in Python to build model to predict average rating with 54% accuracy.
* Discovered most probable terms occurring in 5-star Yelp reviews and their distribution over 50 million words.