RAMKISHAN PANTHENA

ramkishan.panthena@gmail.com | 857-260-8834 | Boston, MA | ramkishanpanthena | RamkishanPanthena

Education

Northeastern University, Boston, MA — August 2019
Master of Science in Data Science, with Thesis
Thesis: **Word-vector Regularization for text classification algorithms (ML, NLP)**
Relevant Courses: Machine Learning, NLP, Information Retrieval, Algorithms, Parallel Data Processing in MapReduce/Spark
Awards: GE Aviation Hackathon - Won "Most Real Impact Team" Award for solving GE's data challenge problem

University of Mumbai, Don Bosco Institute of Technology (DBIT), India — May 2009
Bachelor of Engineering in Electronics and Telecommunication with First Class Honors

Technical Knowledge

Programming/Scripting Languages:	Python, R, Shell Script, Java, Matlab, Scala, SQL, JavaScript, HTML
ML Tools/Frameworks:	TensorFlow, NLTK, Scikit-Learn, Pandas, Gensim, Spark MLlib, Weka
Big Data/Cloud:	Hadoop, Spark, MapReduce, Azure – Databricks/Storage, AWS – EMR/EC2/S3, Heroku
Databases:	Teradata 15X/14X/13X, Elasticsearch, Oracle, Netezza, SQL Server, MongoDB, SQLite
Visualizations:	Matplotlib, GGplot2, D3, SVG, Plotly, Seaborn, NetworkX, Tensorboard, Tableau
NLP algorithms implemented:	Trigram Language Modeling, Part-of-speech tagging, Brown Clustering, Naïve Bayes
IR algorithms implemented:	Document Indexer, Web Crawler, PageRank, Vector space and Language models
Certifications:	Azure Data Science Associate, Azure Fundamentals

Research and Professional Experience

Grantham Mayo Van Otterloo & Co., Boston, MA — Sep 2019 – Present
Data Science/Machine Learning Engineer

- Deployed n-gram topic model in production that had more interpretable topics over unigrams. These n-grams were generated at scale and carefully pruned based on their Part-of-speech tags and Pointwise mutual information scores
- Implemented NLP papers and built a library of reusable ML/NLP tools in PySpark that could manage large workloads
- Developed an in-house financial sentiment engine by fine-tuning a pre-trained FinBERT model for sentiment classification, improving performance of the sentiment engine from 57% to 88%

Earnings Call Transcripts to detect Profit Warnings (R&D – NLP, ML, Big Data)

- Research project that parsed the language used in the earnings call using NLP techniques to determine linguistic patterns that systematically lead to higher or lower post-call profit-warnings
- Trained ML model with features like company's historical performance, sentiment expressed on forward-looking statements, self-attribution bias, market-moving topics, call similarity, use of flattery/obfuscation
- Achieved F1-score of 54% after extensive hyper-parameter tuning with GridSearchCV

EDGAR Topic Modelling (Production Deployment – NLP, ML, Big Data)

- Deployed LDA model in production using Azure, PySpark and Databricks to identify topics across SEC EDGAR archive on 10-K annual filings and compared topics of each company to detect other companies which discuss similar topics
- Compared topics generated by Python and Spark LDA models by computing the KL-divergence scores between both models given their topics and weights, reducing final model training time from 3 hours to 4 minutes
- Leveraged Pandas UDFs to run native Python code with PySpark, reducing run-time from over 60 minutes to 8 minutes

Northeastern University, Boston, MA — May 2018 – Aug 2019
Machine Learning/NLP Researcher

- Implemented a novel regularizer using TensorFlow that improves text classification performance when there are useful signals present in synonymous but rare words
- Outperformed logistic regression to achieve over 45% improvement in Set-Accuracy after testing with multiple datasets

Grantham Mayo Van Otterloo & Co., Boston, MA — May 2018 – Dec 2018
Data Science Development Co-op

Start-up Success Prediction (R&D – ML, Interactive-Visualization) GitHub

- Designed and implemented an interactive web-based visualization using D3.js that allows end user to build a ML model to predict start-up success for US companies and visualize the predicted performance of start-ups across each state
- Connected client-side visualization via a REST API to server-side modeling in Python and deployed on Heroku

Fast Time Series (R&D – Big Data)

- Generated and wrote multiple-file HDF5 datasets in parallel using PySpark and Databricks to an Azure Storage Container to augment the current file caching solution, reducing file wait time from 20 minutes to 5 seconds
- Built a REST API with helper functions in MATLAB to cache the files locally from cloud

Teradata, Mumbai, India June 2010 – Dec 2016

Data Engineer

- Implemented data ingestion pipelines to process streaming raw data using Spark. Integrated about 300 million raw records from 10 different data sources and stored in the database reducing data ingestion time by 80%
- Involved in identifying inefficient queries consuming too many system resources and provided solutions to make them run more efficiently. These efforts led to 30% reducing in the system resource consumption saving millions of CPU cycles
- Performed data cleaning, deduplicating and normalizing using Pandas before it will be loaded into the system
- Developed reusable code and libraries in Python to process different file formats like CSV, XML, JSON
- Designed an automated 'Data Migration Tool' using Python and Teradata Utilities to facilitate effortless migration of data from one Teradata environment to another
- Architected the integration solution to move data from Oracle to Teradata. Used Teradata Utilities like FastLoad, MultiLoad, TPump to complete the migration in the expected time frame
- **Awards:** Teradata ADC Employee of the Quarter (Q4 2014), ADC Project Long Service Award (June 2012 and June 2014)

Academic Projects

Northeastern University, Boston, MA

End-to-end Neural Architecture for Reading Comprehension GitHub

- Implemented a match-LSTM and Answer-Pointer model in TensorFlow for machine comprehension that predicts the start and end position of an answer span, given a question and passage
- Achieved an Exact Match score of 63% on the SQuAD Dataset

Vertical Search Engine

- Implemented a web crawler in Python that crawled over 20,000 web links to construct a collection of documents focused on a particular topic while adhering to politeness policy of the websites
- Stored the data in Elasticsearch and tied it up with UI to allow searching through crawled content
- Implemented PageRank algorithm to compute the PageRank for every page in the crawl

Foreground-Background Pixel Classification for Brain scans GitHub

- Classified a highly unbalanced dataset with over 99.46% data belonging to one class
- Improved training data diversity by performing rotations and mirroring in the X-Y plane; and reduced class imbalance by under-sampling background records
- Achieved a final accuracy of 99.74% using Random Forests and Boosted Trees

Movie Recommender System

- Built a system that examined the ratings of movies provided by users to predict movies that a user might like
- Used K-NN, K-Means, Matrix Factorization, Perceptron for comparison and achieved F1-score of 69.12%

Sentence Generation with Language Modelling GitHub

- Implemented trigram language model with unknown word handling and smoothing
- Generated plausible sentences and achieved perplexity of 79.8 on the test set using interpolation smoothing

Sentiment Analysis of IMDB Movie Reviews

- Detected the overall sentiment of a text review and achieved an F1-score of 81% on the test set
- Naïve Bayes, Multilayer Perceptron and LSTM using word vector embeddings were used for performance comparison