



# RAMKISHAN PANTHENA

✉ panthena.r@husky.neu.edu | ☎ 857-260-8834 | Boston, MA |  ramkishanpanthena |  RamkishanPanthena

## Education

Northeastern University, Boston, MA

May 2019

**Master of Science in Data Science**

**Relevant Courses:** Machine Learning, NLP, Information Retrieval, Algorithms, Parallel Data Processing in MapReduce/Spark

**Awards:** GE Aviation Hackathon - Won "Best Real Impact Team" Award for solving GE's data challenge problem

Don Bosco Institute of Technology (DBIT), University of Mumbai, 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

## Research and Professional Experience

Northeastern University, Boston, MA

May 2018 – Present

**MS Thesis: Word-vector Regularization for text classification algorithms (R&D – ML, NLP)**

- Aim to improve classification performance in cases where there are useful signals present in synonymous but rare words
- Propose a novel regularizer using TensorFlow that assigns similar weights to words with nearly the same meaning
- Generate word-vectors for unigrams/bigrams and train the model to perform multi-class/multi-label classification

Grantham Mayo Van Otterloo & Co., Boston, MA

May 2018 – Dec 2018

**Data Science Development Co-op**

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

- Research project that parses 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 last quarter/last year performance, Sentiment expressed on forward-looking statements, Fog Index, Vocab Diversity on both the prepared management remarks and analyst QA section
- Achieve 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

**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
- Developed a medium through which user can visualize model performance through an ROC curve; and allowed comparison between models with different hyper-parameters by juxtaposition of several ROC curves
- 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 India Pvt. Ltd, Mumbai, India

June 2010 – Dec 2016

**Technical Consultant**

- Conducted statistical analysis on various systems, to ensure efficient system health, and resolved anomalies within the data in a timely manner, improving system performance by 30%
- Designed an automated 'Data Migration Tool' using Linux Shell Scripting and Teradata Utilities to facilitate effortless migration of data from one Teradata environment to another
- Developed Stored Procedures and User Defined Functions in SQL, C, Python to implement user specific requirements
- Conducted research project on loading and retrieving unstructured data (images/photos) using BLOB, JSON data types
- Created and optimized complex ETL/ELT jobs using Teradata utilities, Teradata Parallel Transport(TPT) and Viewpoint
- Designed ETL for implementation of Slowly Changing Dimension(SCD) Type1/Type2 loads
- **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