Wang Jin *(jin.w@husky.neu.edu)*

(617)369-2483 • Windsor Street, Cambridge, MA, 02139 • linkedin.com/in/wang-jin • github.com/jinwangjoshua

TECHNICAL SKILLS

Open Source: **Python, SQL, R**; Scikit-learn, Keras, Tensorflow, Spark-MLib, NLTK, Spacy, Gensim, TPOT, H2o, Pandas, Numpy
ETL/Database: Airflow, PostgreSQL, MySQL, DB2, MSSQL, MongoDB, Redis, Hadoop/HDFS/ MapReduce, Hive, Spark
ML/Statistics: GLM, SVM, Naïve Bayes, DT, RF, XGBoost, Clustering, DP, Statistical Inference, Hypothesis Test, Time Series
Other Tools/Skills: Linux/Shell, Git/Bitbucket, Docker, Tableau, Jupyter, Web Scraper, A/B test, DoubleClick Certificate

EDUCATION

Northeastern University, Boston, MA ***Dec. 2018***
Data Analytics Engineering, Master of Science, GPA: 3.5/4.0
Courses: Algorithms | Data Mining in Eng. | Statistical Methods in Engineering | Advances in Data Science | Operation Research
Awards*: Won Third Place in Wayfair Datathon on Nov. 3, 2018.*
Nanjing Univ. of Aeronautics and Astronautics, Nanjing, China ***Jul. 2016***
Aeronautical Engineering, Bachelor of Engineering, GPA: 85/100

WORK EXPERIENCE

Data Scientist, Digital Remedy, New York City, NY ***Jan. 2018-Jun. 2018***
Social Media Posts Popularity Prediction

- Predict the popularity of new social media posts using machine learning models and Automated machine learning
- Translated real business targets to proper data mining tasks and designed feature collection schema
- Applied algorithms of *GBDT, Random Forest, Logistic and Lasso Regression, KNN, etc.*
- Designed ETL (Data Extraction, Transformation, Loading) with DoubleClick, Gosquared and Facebook APIs and MySQL

Text Data Mining of Internet Trending

- Constructed a text mining system to attract website visiting traffic and improve ads impressions and profit
- Cleaned text data on a scale of 100k rows and created specific stop words database through *Python and NLTK*
- Used *Fuzzy Matching* method based on similarity distances *(Levenshtein, Jaro-Winkler, etc.)* to track internet trending
- Evaluated model effectiveness through the recall of Top K results and deployed it on the NoSQL Database (MongoDB)

Anomaly Detection of Traffic Trending

- Completed an *Anomaly Detection system* to monitor ads impressions trending using *Time Series Method (Holt-Winter)*

RESEARCH EXPERIENCE

Opinion Extraction(NLP) on Ecommerce Proview Reviews ***Aug. 2018-Present***

- Generated the most frequent opinion phrases by *POS tagging* and extraction rules; completed by *Spacy, Gensim*
- Group opinion phrases based on semantic similarities based on *Word2vec*

PROJECT EXPERIENCE

Alibaba Advertising Algorithm Contest – Post Click Conversion Rate (CVR) Prediction ***Mar. 2018-Apr. 2018***

- Implemented machine learning models to predict the conversion rate after the users' clicks on search recommendations
- Trained *Xgboost and Lighgbm* model on around-500,000 rows of historical user behaviors data
- Completed data cleaning according to business logics; Visualized and analyzed user behaviors
- Created features to improve model, such as *time difference, statistical quantity in time window and Jaccard index*
- Called automatic hyperparameter tuning (*Simulated Annealing and Random Search* methods) for better parameters
- Predict of probability of conversion and evaluated performance by *Log-loss metric*

Image Recognition - Deep learning ***Oct. 2017-Dec. 2017***

- Realized Face Recognition by applying a *Convolutional Neural Network (CNN)* classification model
- Called *TensorFlow* to generate a deep neural network classifier, and achieved a classification accuracy of 97.9%
- Captured and pre-processed image data with *Opencv and Dlib*
- Used dropout and different data argumentation methods to regularize and avoid overfitting

Time series: Stocks Price Forecasting Model ***Oct. 2017- Dec. 2017***

- Predicted future prices based on historical stock data by applying several Time Series Algorithms
- Collected and cleaned stock data from Yahoo Finance; Completed *Explorable Data Analysis and Data Visualization*
- Used *3-Exponential Smoothing and Autoregressive Integrated Moving Average (ARIMA) Model*, with Python and R