Joseph Mok

Email: [*hojinmok@gmail.com*](mailto:hojinmok@gmail.com) GitHub: [*github.com/hjmok*](https://github.com/hjmok)

Portfolio: [*hjmok.github.io/josephmok\_portfolio*](https://hjmok.github.io/josephmok_portfolio) LinkedIn: [*linkedin.com/in/hojinjosephmok*](https://www.linkedin.com/in/hojinjosephmok/)

# Skills

* Proficiency in utilizing TensorFlow/Keras, PyTorch, Scikit-Learn, OpenCV, NumPy, and Pandas libraries in Python
* Programmed various machine learning models including linear/logistic regression, CNN, RNN, NLP models, etc.
* Well versed in data preprocessing and feature engineering in preparation of machine learning model training
* Experienced with SCADA and PLC software packages, including Allen Bradley and Ignition Automation Software
* Experienced integrating MySQL, Python scripting, and Ignition for capturing SCADA trend data and data analysis
* Strong cross-functional project management experience from leading various multidisciplinary projects

# Professional Experience

Yaya Foods Corp. *Toronto, ON*

**Automation Engineer***,*July 2019 – Present

* Lead the automation design for several SCADA systems for beverage manufacturing clients by integrating Rockwell Automation PLCs/HMIs, Ignition Automation Software, MySQL, and ethernet IP networks
* Developed data mining process by storing transmitter data from PLCs to MySQL database using Ignition SCADA packages. Developed client applications within Ignition for data analysis (heavy Python scripting)
* Gained strong troubleshooting skills for hardware and software to determine root cause of production halts
* Sourced transmitters, motors, VFDs, valves, and necessary hardware for new automation systems

Apple Inc. *Cupertino, CA*

**Recycling R&D Engineer Intern***,*September 2018 – April 2019

* Managed development of recycling systems for Apple products. Involved heavy coordination with vendors and internal Product Design teams to organize documentation and deadlines
* Developed optimal cycle time processes for de-manufacturing of various Apple products. Resulted in a 90% increase in the Unit-per-Hour output compared to the current manual methods
* Designed semi-autonomous prototypes to showcase cycle time and disassembly improvements. Designs continuously integrated feedback to improve operator ergonomics and meet California waste regulations

# Projects

The following are sample projects. To see all projects, please visit: [*https://hjmok.github.io/josephmok\_portfolio*](https://hjmok.github.io/josephmok_portfolio)

**Breast Cancer Malignant or Benign Diagnosis** [*https://hjmok.github.io/josephmok\_portfolio/#/BC*](https://hjmok.github.io/josephmok_portfolio/#/BC)

* Applied Logistic Regression and K-Nearest Neighbour models to a Breast cancer classification dataset with 32 columns of patient data, including a label which determine if the cancer cells were malignant or benign
* Used Scikit-Learn’s LogisticRegression and KNeighborsClassifier classes
* Logistic Regression model achieved 98% accuracy and K-Nearest Neighbour achieved 97% accuracy

**TFIDF Text Classification** [*https://hjmok.github.io/josephmok\_portfolio/#/TFE*](https://hjmok.github.io/josephmok_portfolio/#/TFE)

* Created a supervised learning model to classify Positive/Negative reviews in an Amazon Reviews dataset and Ham/Spam text messages in an SMS dataset. Data-preprocessing involved removing null rows
* Utilized Scikit-Learn’s TfidfVectorizer to Count Vectorize each unique word in the training set, then applied Term Frequency-Inverse Document Frequency feature extraction to said words. Then used Scikit-Learn’s LinearSVC (Support Vector Classifier) to return the best fit hyperplane to categorize the data.
* Achieved a 98% accuracy on the SMS dataset and 86% accuracy on the Amazon Reviews Dataset

**Stock Prices Prediction** [*https://hjmok.github.io/josephmok\_portfolio/#/StockPriceRNN*](https://hjmok.github.io/josephmok_portfolio/#/StockPriceRNN)

* Created an RNN model on Tensorflow to predict AMD and Google Stock prices by training on daily stock price data from May 2009 to August 2018
* Model utilized 4 LSTM layers and test on a quarterly sequence size and annual sequence size
* Resulting models were able to follow the general trend and scale of the stock prices

**Taxi Fare Prediction ANN**  *https://hjmok.github.io/josephmok\_portfolio/#/Taxi*

* Created ANN Regression and Classification models with PyTorch to predict a taxi fare amount and class
* Performed feature engineering and separated continuous/categorical columns from the data set. Utilized dropout layers and batch normalization to reduce overfitting
* Regression model achieved a MSE of $3.88 (prediction is off by this much on average) and Classification model achieved cross entropy loss of 0.272 (misclassified 27% of the time)

**User Movie Rating Prediction with AutoEncoders** [*https://hjmok.github.io/josephmok\_portfolio/#/AE*](https://hjmok.github.io/josephmok_portfolio/#/AE)

* Created a Stacked AutoEncoder model to predict what rating a user will give to a film. Model used 3 hidden layers with 32, 16, and 32 nodes for the first, second, and third hidden layers respectively.
* Trained on a Grouplens dataset with 1 million rows of move ratings from 6040 users across 3952 rows
* Achieved a train loss of 0.808 and test loss of 0.896, meaning every prediction is off by ±1 star

# Education

University of Waterloo,

**Bachelor of Applied Science, Honours Mechanical Engineering,** Graduated June 2019

*Extras:*

**Object Detection with Single Shot MultiBox Detection**  [*hjmok.github.io/josephmok\_portfolio/#/SSD*](https://hjmok.github.io/josephmok_portfolio/#/SSD)

* Implemented a pre-trained SSD model using PyTorch to perform Object Detection on videos
* Created a Detect Function to take input video frames and convert them into Torch variables that the SSD model accept. Then used OpenCV to apply red rectangles around the detected objects. Video in link above

**Face Detection with Haar Cascades**  *hjmok.github.io/josephmok\_portfolio/#/FD*

* Used OpenCV to load pre-trained Haar Cascades for the face, eyes, and smile
* Created a Detect function which applies rectangles around the regions of interest detected by the Haar Cascade to the input video frames. Performed face recognition using the user’s webcam. Video in link above

**Taxi Fare Prediction ANN**  *hjmok.github.io/josephmok\_portfolio/#/Taxi*

* Created ANN Regression and Classification models with PyTorch to predict a taxi fare amount and class
* Performed feature engineering and separated continuous/categorical columns from the data set. Utilized dropout layers and batch normalization to reduce overfitting
* Regression model achieved a MSE of $3.88 (prediction is off by this much on average) and Classification model achieved cross entropy loss of 0.272 (misclassified 27% of the time)