

# ANDREW J. ZHOU

Ridgefield, WA | 646-460-2231 | [ajz55@uw.edu](mailto:ajz55@uw.edu)  
<https://github.com/azhou5211> | <https://www.linkedin.com/in/andrew-j-zhou/>

## EDUCATION

**UNIVERSITY OF WASHINGTON** | Seattle, WA

*Anticipated graduation March 2022*

**Master of Science in Data Science** | GPA: 3.97

**Honors/Awards:** Winner of Best Prediction Model for UW Data Science Hackathon 2020

**Relevant Coursework:** Statistics and Probability, Data Visualization, Machine Learning for Big Data, Software Design

**RUTGERS UNIVERSITY** | New Brunswick, NJ

May 2019

**Bachelor of Science in Computer Science** | GPA: 3.79

**Bachelor of Arts in Statistics** | GPA: 3.62

**Honors/Awards:** Dean's List, Computer Science Departmental High Honors, Statistics Departmental Honors

**Relevant Coursework:** Information and Data Management, Computing and Graphics in Statistics, Artificial Intelligence

## TECHNICAL SKILLS

**Coding Languages:** Python, Java, C, R, SQL, HTML, SAS, JSP

**Coding Packages:** Scikit-Learn, NumPy, Matplotlib, Pandas, Flask, TensorFlow

**Operating Systems:** Windows, Linux, Mac OS

**Other Tools:** Tableau, Microsoft Office, Adobe Photoshop, Adobe InDesign

**Languages:** Native English, Chinese

## WORK EXPERIENCE

**The Clorox Company, Data Science Intern** | Pleasanton, California

June 2021 – August 2021

- Developed end-to-end Multi-Touch Attribution Model to attribute credit to marketing campaigns for a customer purchase
- Designed modules to query data, clean the data, model the data, and generate a visualization for the user
- Saved marketing spending on underperforming marketing campaigns while maintaining optimal customer purchase
- Designed and developed a trend anomaly diagnostics system which assisted in understanding how and why an anomaly occurred. This enabled users to interpret and take action when facing a detected trend anomaly alert

**Inner Mongolia Sunnergy Co., Data Science Intern** | Hohhot, China

May 2018 – August 2018

- Collaborated with factory engineers to design an experiment to collect data about factory machines and output
- Generated regression models with data; analyzed and evaluated the models for accuracy
- Discovered which machine factors impact output the most, and how to modify those machine factors
- Achieved 4% faster machine output by modifying impactful machine factors

## PROJECTS

**Recipeat** (Python, Jinja, PostgreSQL, Firebase, HTML)

Spring 2021

Developed a website for users to search recipes with given ingredients and nutritional constraints

- Oversaw the project as Team Co-Leader and delegated assignments to group of 5 team members
- Designed HTML forms for users to register accounts, search for recipes, and compare recipes
- Integrated the user interface with flask to communicate with back-end python modules
- Wrote back-end Python modules that connected to the PostgreSQL database to insert, search, update, and delete data
- Implemented unit tests for back-end modules and integrated Travis CI with GitHub to ensure pushed code is reliable
- Achieved an end-to-end website for users to find recipes and compare recipes with given ingredients and nutritional constraints

**COVID-19 and Government Policies** (Python, SAS JMP)

Summer 2020

Find which government policies were the most effective at slowing down COVID-19 cases

- Searched and scraped data online regarding COVID-19 cases and government policies implemented in the United States
- Cleaned the data into a usable format for machine learning algorithms by using Pandas package from Python
- Analyzed government policies from generated auto-regression time series and SAS JMP graphic visualizations and removed non-impactful government policies from the regression
- Discovered the most impactful government policies were mask mandates and gathering restrictions of any size

**Digit and Facial Classification** (Python)

Fall 2018

Implemented Naïve Bayes, Perceptron, and MIRA algorithms to identify values of handwritten digits and to detect faces in photos

- Split the data set into three categories: training set, validation set, and test set
- Developed code to train all three algorithms on the training data set
- Analyzed the three algorithms predicting abilities on testing set; discovered Naïve Bayes worked best with an 88% prediction rate