

# Justin Lai

[justinlai@gmail.com](mailto:justinlai@gmail.com) | [linkedin.com/in/lai-justin](https://www.linkedin.com/in/lai-justin) | [github.com/Lai-Justin](https://github.com/Lai-Justin) | [lai-justin.github.io](https://lai-justin.github.io)

## EDUCATION

**University of California, Santa Barbara**

September 2019 – June 2023

*Bachelor of Science in Computer Science, College of Engineering*

*GPA: 3.61*

## EXPERIENCE

**Machine Learning Engineer Intern**

June 2022 – August 2022

*Sermonis.ai*

*Berlin, Germany*

- Developed web scraper to extract text from online news articles, enabling efficient data collection
- Cleaned, labeled, and converted the text into JSONs, optimizing data quality and compatibility with NLP models
- Utilized the Huggingface library to tokenize the text data for streamlined preprocessing and feature extraction
- Assisted with deploying Elasticsearch to index and store the tokenized data for efficient querying and retrieval
- Leveraged Elasticsearch to effectively store and search the tokenized data, enabling fast and accurate information retrieval for downstream tasks

**Frontend Web Developer Intern**

June 2018 – August 2018

*University of California, San Francisco*

*San Francisco, California*

- Improved UI features in The Ilios Project's medical curriculum manager, such as task lists, reminders, and toolbars
- Revised the IndustryDocumentLibrary's UI, enhancing accessibility, viewing experience, and content delivery
- Enhanced text scraping in IndustryDocumentLibrary for efficient metadata extraction from legal documents
- Processed and redacted medical documents for online publication, ensuring confidentiality and compliance
- Contributed to UI/UX discussions, actively suggesting enhancements in meetings for continuous improvement
- Performed QA and unit and integration testing for both projects using JUnit, Selenium, and Mockito

## PROJECTS

**CIFAR-10 Convolutional Neural Network Image Classifier** | *Python, TensorFlow, Keras, Scikit-Learn, Matplotlib*

- Developed a deep learning model for classifying 32x32 colored images of objects, based on the CIFAR-10 dataset
- Trained the CNN model using a dataset of 50,000 images for enhanced learning and generalization
- Utilized One-Hot Encoding to convert categorical labels into binary representations for improved training
- Implemented a multi-layer CNN to capture spatial dependencies in image data, enhancing classification accuracy
- Tuned hyperparameters by adding batch normalization, pooling, and dropout layers to ensure robustness
- Visualized model training and outputs to analyze and better understand CNN feature extraction and learning

**Four Year Schedule Planner** | *JavaScript, Python, React.js, Node.js, Express.js, Mongoose*

- Developed a MERN-based course scheduling web app for UC Santa Barbara students to plan four-year schedules
- Implemented a web scraper to gather accurate and up-to-date course IDs, titles, and grade distributions
- Created an intuitive user interface to add and manage university-specific courses in personalized schedule calendars
- Utilized Mongoose and React.js for efficient data storage and dynamic, responsive user experience
- Collaborated with professors and students to gather feedback, iterate on the website's features, and ensure its relevance and usability

**Athlete Fitness Naïve Bayes' Classifier** | *Python, Numpy, Pandas*

- Developed an AI binary classifier using Naïve Bayes algorithm to assess athlete fitness based on 11 health criteria
- Trained the classifier using a dataset of 9000 entries containing numerical values representing various health factors
- Implemented conditional probability calculations to generate accurate fitness predictions for each criterion
- Fine-tuned the classifier's performance through rigorous testing, achieving an accuracy of 87% on the test dataset

**Temperature-based Automated Window Control Software** | *Python, GPIOZero, PostgreSQL, Flask, Request*

- Developed an automated window control software using a Raspberry Pi, temperature sensors, and motors
- Integrated geolocation and NationalWeatherService APIs to fetch outdoor temperature and other weather data
- Utilized PostgreSQL to store indoor temperature data for machine learning analysis to improve algorithm accuracy
- Created a Flask server to display real-time indoor/outdoor temperatures and window status

## TECHNICAL SKILLS

**Languages:** Python, C++, C, R, OCaml, JavaScript, TypeScript, Java, SQL, MIPS, C#, HTML5, CSS, LaTeX

**Tools:** Git, VS Code, Bash, Unix/Linux, Anaconda, PostgreSQL, Docker, Elastic Search, Kibana, AWS EC2, Kanban

**Frameworks:** Agile, CI/CD, React, Node, MongoDB, Flask, JUnit, RESTful API, Spring Boot, CodeCov, Storybook

**Libraries:** Numpy, Pandas, Seaborn, RSQlite, Scikit-Learn, TensorFlow, PyTorch, HuggingFace, Keras, OpenCV, Spark