

# Vuk Perisic

Berkeley, CA \* (714)746-8177 \* [v\\_perisic@berkeley.edu](mailto:v_perisic@berkeley.edu)

---

## Education

**University of California, Berkeley**  
Bachelor's Degree in CS

**Expected Graduation: May 2024**  
**GPA: 3.65**

**Relevant Coursework:** Data Structures and Programming Methodology (**Data structures and algorithms in Java**), Structure and Interpretation of Computer Programs (**Python programming**), Foundations of Data Science (**A/B testing, bootstrapping, k-means classification, introductory MySQL**), Computation Structures in Data Science (**Python programming**), Discrete Math and Probability, Designing Information Devices and Systems (**Linear algebra**)

---

## Skills

**Languages:** Python, Java, JavaScript, HTML/CSS, MySQL

**Libraries:** Pandas, Scikit-Learn, XGBoost, Numpy, Pytesseract

---

## Projects

### eBay University Machine Learning Competition

**Fall 2021**

- Developed machine learning model to predict delivery date of a package based on past seller shipping information using XGBoost and scikit-learn libraries
- Prepared real-world dataset containing 2.5 million entries, missing values, and incorrect data in some cases for analysis using Pandas
- Placed on leaderboards with team of 4

### BeachHacks

**Spring 2021**

- Won 1st Place and Best Interdisciplinary Hack in BeachHacks competition
- Worked with team of 5 to develop Leetcards, an app that takes in a picture of the user's code as input, runs it through test cases, and returns whether it is a valid solution to a given problem
- Wrote back end code that applied Pytesseract library to convert a picture of Python code to text, addressed formatting errors, and ran the user's code through test cases

### Gitlet

**Summer 2021**

- Replicated core functionality of Git version-control system
- Implemented features like reading file contents, making commits, viewing and restoring earlier versions of files, and merging changes between branches

### ArrayDeque

**Summer 2021**

- Programmed an Array Deque in Java from scratch using built-in array data structure
- Utilized circular array structure and minimum usage factor of 25% to implement constant-time add, remove, get, and size operations

### Scheme Interpreter

**Spring 2021**

- Built basic interpreter for the Scheme programming language in Python
- Parsed Scheme inputs in the form of strings and returned results of the evaluated expression
- Handled user-defined functions, logical operators, variables, and built-in procedures

**In Progress:** Currently taking Udemy courses on React.JS and HTML/CSS