

- SMARTCORE -
AN ADVANCED RUST MACHINE LEARNING LIBRARY

CSCI 3656 - COMMUNITY ANALYSIS

JAREK REYNOLDS

04 APRIL 2023

HISTORY & OVERVIEW

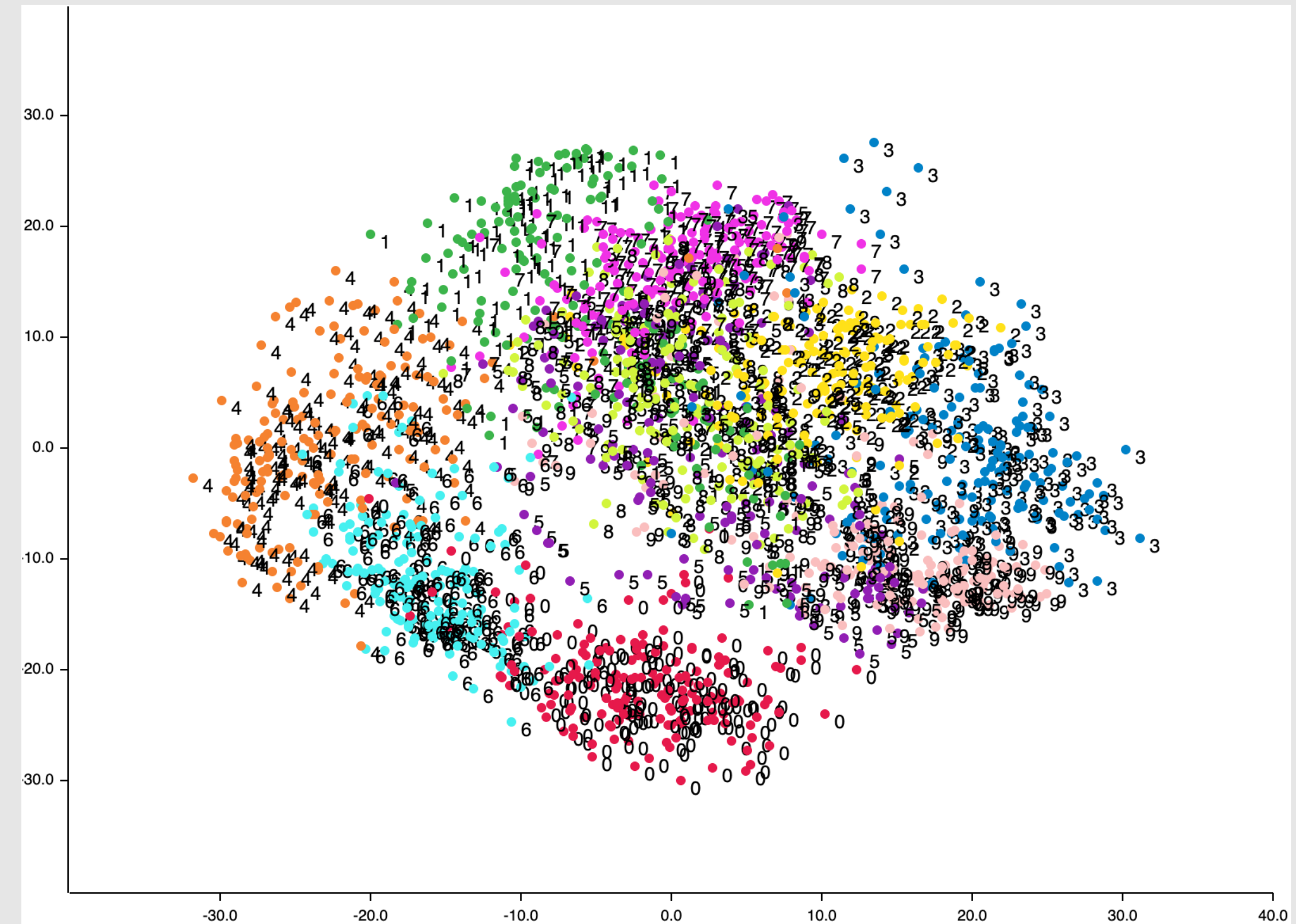
- ▶ Rust is a programming language roughly seven years old. It is a multi-paradigm, high-level, general-purpose programming language emphasizing performance, memory safety, and concurrency.
- ▶ Last year Rust became the second language after C supported in Linux kernel development.
- ▶ SmartCore is a machine learning and numerical computing library for Rust. SmartCore provides tools for linear algebra, numerical computing, and optimization. SmartCore is touted as the most advanced ML library for Rust (SmartCore).

DIMENSIONALITY REDUCTION - PCA EXAMPLE - DIGITS DATASET

```
use smartcore::dataset::*;  
// DenseMatrix wrapper around Vec  
use smartcore::linalg::naive::dense_matrix::DenseMatrix;  
// PCA  
use smartcore::decomposition::pca::{PCA, PCAParameters};  
// Load dataset  
let digits_data = digits::load_dataset();  
// Transform dataset into a NxM matrix  
let x = DenseMatrix::from_array(  
    digits_data.num_samples,  
    digits_data.num_features,  
    &digits_data.data,  
);  
// These are our target class labels  
let labels = digits_data.target;  
// Fit PCA to digits dataset  
let pca = PCA::fit(&x, PCAParameters::default().with_n_components(2)).unwrap();  
// Reduce dimensionality of X to 2 principal components  
let x_transformed = pca.transform(&x).unwrap();
```

PCA Example Code - Digits Dataset (SmartCore)

Principal component analysis (PCA) is an approach to dimensionality reduction. PCA is a popular approach for deriving a low-dimensional set of features from a large set of variables (Rust).



First two principal components of the Digits dataset (SmartCore).

QUESTIONS

- ▶ How does the performance of Rust/SmartCore compare to a more mature programming language and library like Julia/LinearAlgebra or Python/SciPy/NumPy?
- ▶ How does accuracy and error compare between the programming languages and libraries when solving easy and difficult versions of the same problems?
- ▶ Is Rust/SmartCore more performant despite most of NumPy and SciPy being written in C and Fortran?

EXPERIMENTS

- ▶ For this project, I would like to focus on numerical methods for linear algebra covered in class.
- ▶ More specifically, I would like to explore the performance, error, and accuracy of the SVD, Cholesky Decomposition, and QR Factorization methods.
- ▶ Last, I would like to conduct a comparison analysis of Rust/SmartCore against Julia/LinearAlgebra and Python/NumPy using the same problems and see how they are affected as the problems increase in difficulty, dimensionality, and complexity.

REFERENCES

- ▶ Rust. (n.d.). [–][src]module smartcore::decomposition. smartcore::decomposition - Rust. Retrieved April 3, 2023, from <https://docs.rs/smartcore/0.2.0/smartcore/decomposition/index.html>
- ▶ SmartCore. (n.d.). Unsupervised learning. SmartCore. Retrieved April 3, 2023, from https://smartcorelib.org/user_guide/unsupervised.html#dimensionality-reduction