SCIKIT-LEARN RECREATION AND IMPLEMENTATION PROJECT

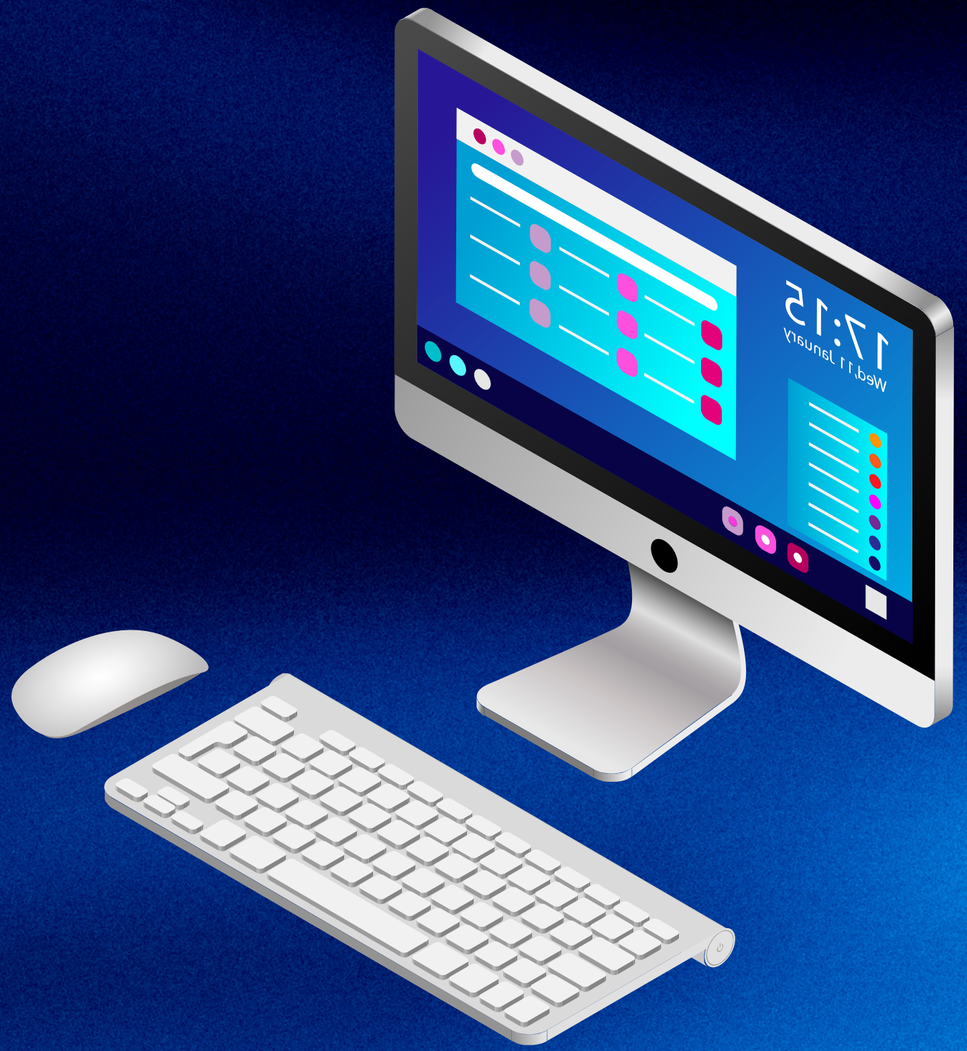
GROUP-5
SALMAN SHAFI-2211386042
JARINAH TASNIM-2121026642
JESMIN AKTER-2111382642
SAZID RAHMAN BHUIYAN-2131267642

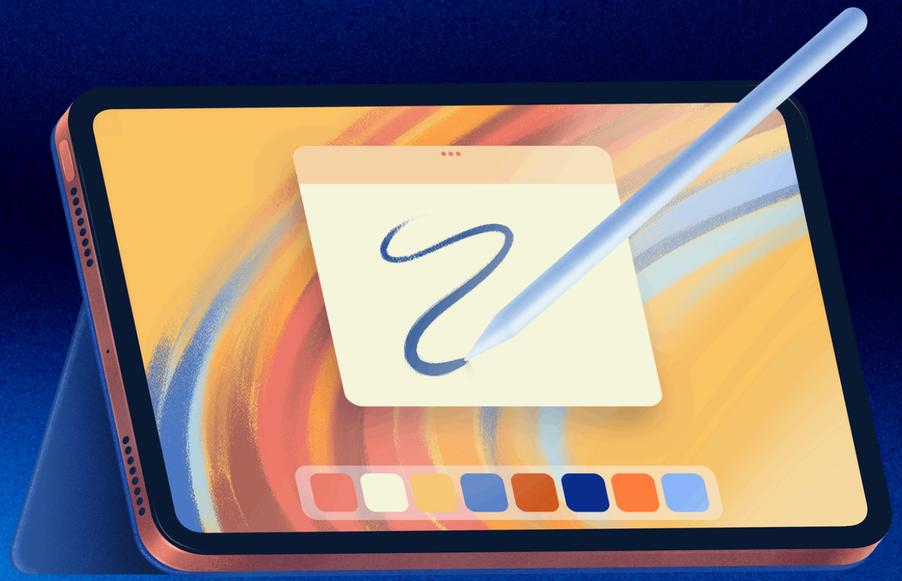
PROJECT OVERVIEW

What We're Building

- Complete recreation of Scikit-Learn library functionality
- Command-line accessible machine learning toolkit
- Two-phase implementation approach
- Educational focus on algorithm understanding
- Local machine execution without external dependencies

Goal: Build a comprehensive ML library from scratch that matches scikit-learn capabilities





PROBLEM STATEMENT

Why This Project Matters

- **Students often use ML libraries as "black boxes"**
- **Limited understanding of underlying algorithms**
- **Need for hands-on implementation experience**
- **Gap between theory and practical implementation**
- **Command-line proficiency essential for developers**

Solution: **Build from ground up to understand every component**

PROJECT PHASES

Phase 1:
Implementation of Scikit-Learn Examples

- Classification algorithms(SVM, RandomForest, Logistic Regression)
- Regression techniques (Linear, Ridge, Lasso)
- Clustering methods (K-Means, DBSCAN)
- Dimensionality reduction (PCA, t-SNE)
- Model selection and validation
- Data preprocessing utilities

Phase 2:
Complete Recreation

- Custom Library architecture
- Algorithm implementation from scratch
- Command-line interface development

TECHNICAL SCOPE- ALGORITHMS

Classification

- **Support Vector Machines, Random Forest, Logistic Regression, Decision Trees, Naive Bayes**

Clustering:

- **K-Means, DBSCAN, Hierarchical Clustering, Gaussian Mixture Models**

Regression:

- **Linear Regression, Ridge Regression, Lasso Regression, Polynomial Regression**

Dimensionality Reduction:

- **Principal Component Analysis (PCA), t-SNE, Linear Discriminant Analysis (LDA)**

TECHNICAL SCOPE - UTILITIES

Model Selection:

- Cross-validation techniques
- Grid Search optimization
- Train-test functionality
- Performance metrics

Preprocessing:

- Data scaling and normalization
- Feature encoding (categorical variables)
- Missing value handling
- Feature selection methods

Command-Line Interface:

- Interactive parameter input
- Batch processing capabilities
- Result visualization options

IMPLEMENTATION STRATEGY

DEVELOPMENT APPROACH

- Study existing scikit-learn documentation
- Implement mathematical foundations first
- Build modular, reusable components
- Extensive testing against known datasets
- Performance benchmarking

TOOLS AND TECHNOLOGIES

- Python programming language
- NumPy for numerical computations
- Command-line argument parsing
- Local file system integration

EXPECTED DELIVERABLES

Phase 1 Deliverables:

- Working examples of all major algorithm categories
- Command-prompt executable scripts
- Documentation for each implementation
- Test results and validation

Phase 2 Deliverables:

- Complete custom ML library
- Comprehensive command-line interface
- User documentation and guides
- Performance comparison reports
- Educational materials explaining implementations

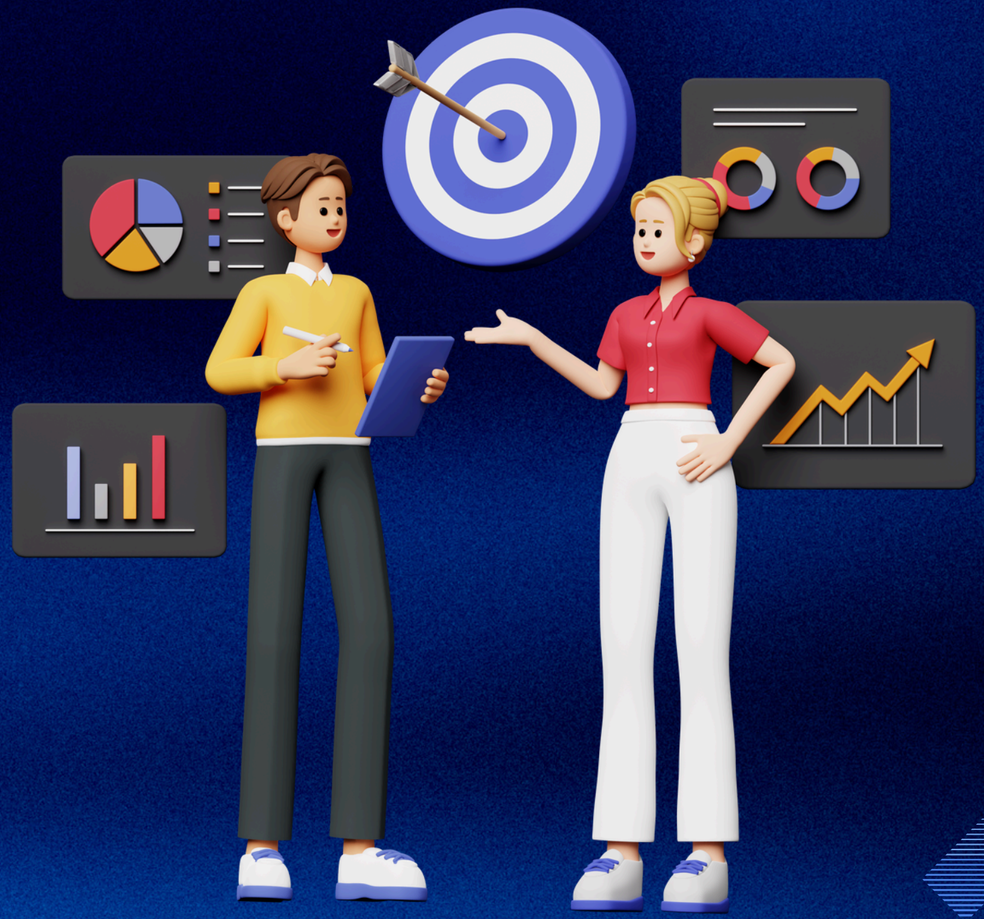
SUCCESS METRICS

Technical Success:

- All algorithms produce accurate results matching scikit-learn
- Command-line interface provides full functionality
- Performance meets acceptable benchmarks
- Code passes comprehensive testing

Educational Success:

- Deep understanding of ML algorithm internals
- Ability to explain implementation details
- Practical command-line development skills
- Foundation for advanced ML research



THANK YOU