Machine Learning Project Documentation

Machine Learning Project Documentation: Regression and Classification Analysis

Project Overview

This project performs two distinct machine learning tasks:

1. **Regression Analysis**: Using **Linear Regression & KNN** as regressors on a numerical dataset (insurance costs)
2. **Classification Analysis**: Using **Logistic Regression & K-Means** as classifiers on an image dataset (Fashion-MNIST, 5 classes)
   The project compares different algorithms within each category to demonstrate the effectiveness of various approaches for regression and classification problems.

Part 1: Insurance Cost Prediction (Regression Analysis)

General Information on Dataset

- **Name**: Insurance Cost Prediction Dataset
- **Classes**: Not applicable (Regression Problem)
- **Numeric Features**:
  - age: Age of the insured person
  - sex: Gender (male/female)
  - bmi: Body Mass Index
  - children: Number of children covered
  - smoker: Whether the person smokes (yes/no)
  - region: Geographic region (northeast/southeast/southwest/northwest)
- **Target Variable**: charges (medical insurance costs)
- **Total Samples**: 1338 samples
- **Feature Dimensions**: 8 features after preprocessing (age, bmi, children, and 5 one-hot encoded categorical variables)
- **Training Samples**: 1070 (80% of data)
- **Testing Samples**: 268 (20% of data)

Implementation Details

Feature Extraction Phase

- **Features Extracted**: 8 features total after preprocessing
- **Feature Names**:
  - age (numeric)
  - bmi (numeric)
  - children (numeric)
  - sex_male (binary, one-hot encoded)
  - smoker_yes (binary, one-hot encoded)
  - region_northeast (binary, one-hot encoded)
  - region_northwest (binary, one-hot encoded)
  - region_southeast (binary, one-hot encoded)

- **Dimension of Resulted Features**: 8-dimensional feature vectors

Cross-Validation

- Cross-validation was not explicitly used in the model evaluation
- Instead, a single train/test split was used (80%/20%)

Hyperparameters Used

- **Linear Regression**: Default parameters (no regularization)
- **KNN Regression**:
  - k=5 (number of nearest neighbors)
- **StandardScaler**: Applied to scale features for KNN model

Model Training Process

- Data was split into training (80%) and testing (20%) sets
- Features were standardized using StandardScaler (especially important for KNN)
- Both models were trained on the same dataset and compared

Results Details

Linear Regression Results (All Features):

- **R² Score**: 0.8069 (on test set)
- **MSE**: 35478021
- **RMSE** : 5956
- **MAE** : 4177
- **Performance Observation**: Good baseline model for regression

KNN Regression Results (All Features):

- **R² Score**: 0.8371 (on test set)
- **MSE**: 29929604
- **RMSE** : 5471
- **MAE** : 3474
- **Performance Observation**: Slightly better than linear regression

Correlation with Charges:

charges	1.000000
smoker_yes	0.787234
age	0.298308
bmi	0.198401
region_southeast	0.073578
children	0.067389
sex_male	0.058044
region_northwest	-0.038695
region_southwest	-0.043637

Observations & Decisions

1. **Strong Correlations**:
   - `smoker_yes`: Very high correlation (> 0.78). This is the most important feature.
   - `age`: Moderate correlation (~0.3).
   - `bmi`: Moderate correlation (~0.2).
2. **Weak Correlations**:
   - `children`: Low correlation (~0.067).
   - `sex_male`: Very low, near zero (~0.057).
   - `region_*`: All region variables have correlations very close to zero (e.g., -0.04, 0.07).

Decision

The relationship between `sex` and `region` with `charges` is **very weak**. Including them might add noise and complexity without adding predictive value.
Action: We will DROP `sex` (including sex_male) and `region` columns. We will keep `children` for now as it influences charges slightly more than the others, but we could experiment with dropping it too.

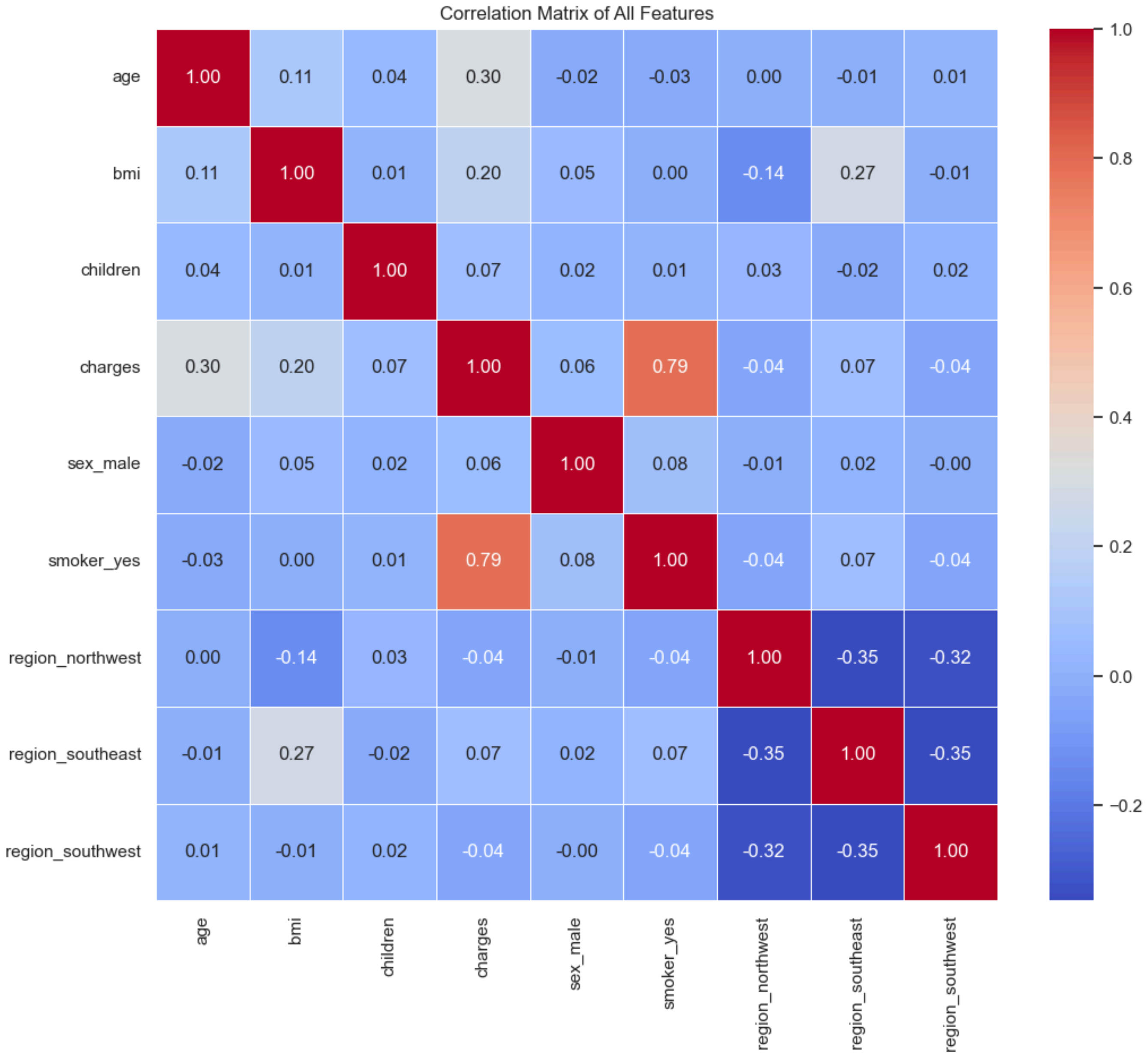
Feature Selection Analysis

The analysis demonstrated that removing weak features (sex and region) had minimal impact on model performance:

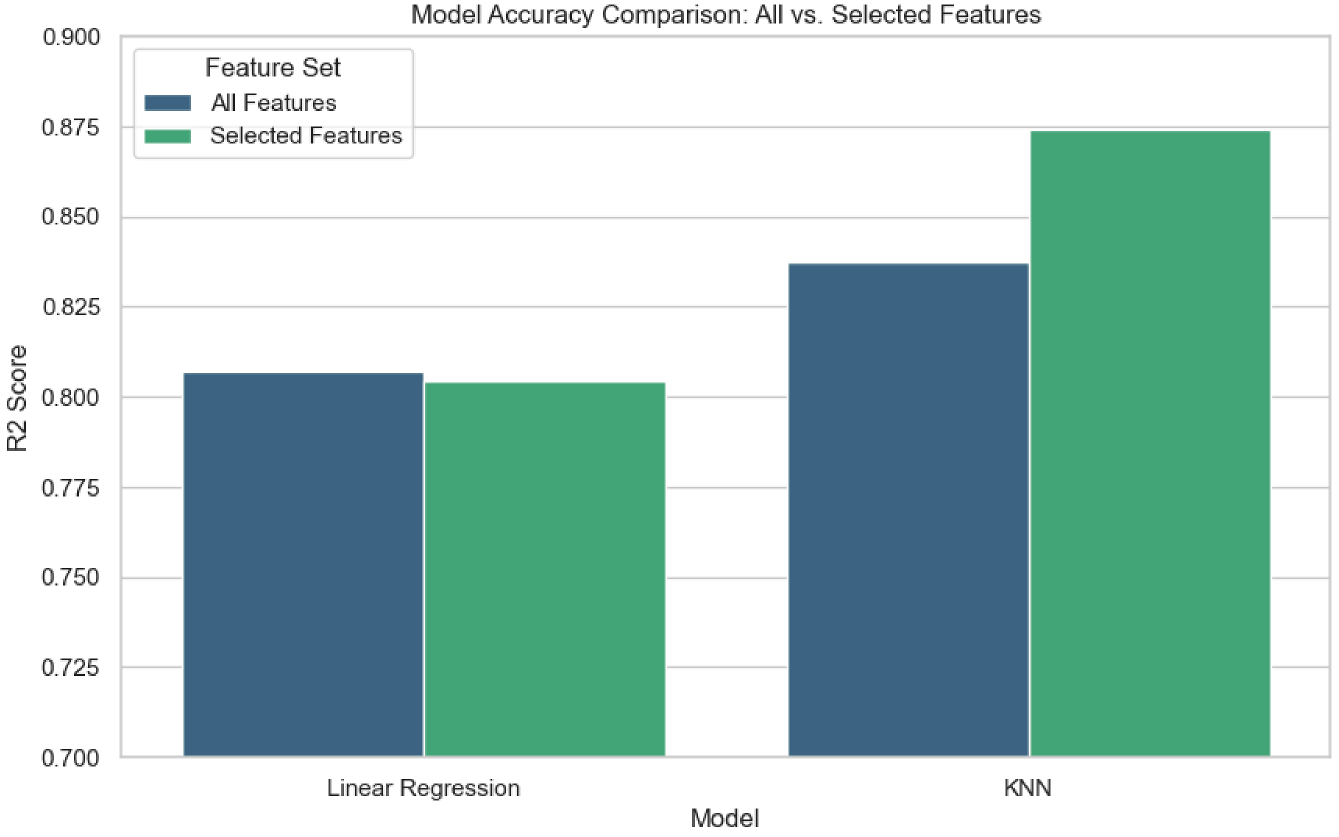
- **With All Features** (8 features):
  - Linear Regression: R² = 0.8069
  - KNN: R² = 0.8371
- **With Selected Features** (4 features - age, bmi, children, smoker):
  - Linear Regression: R² = 0.8046 (minimal decrease)
  - KNN: R² = 0.8739 (significant improvement)

Visualization Results

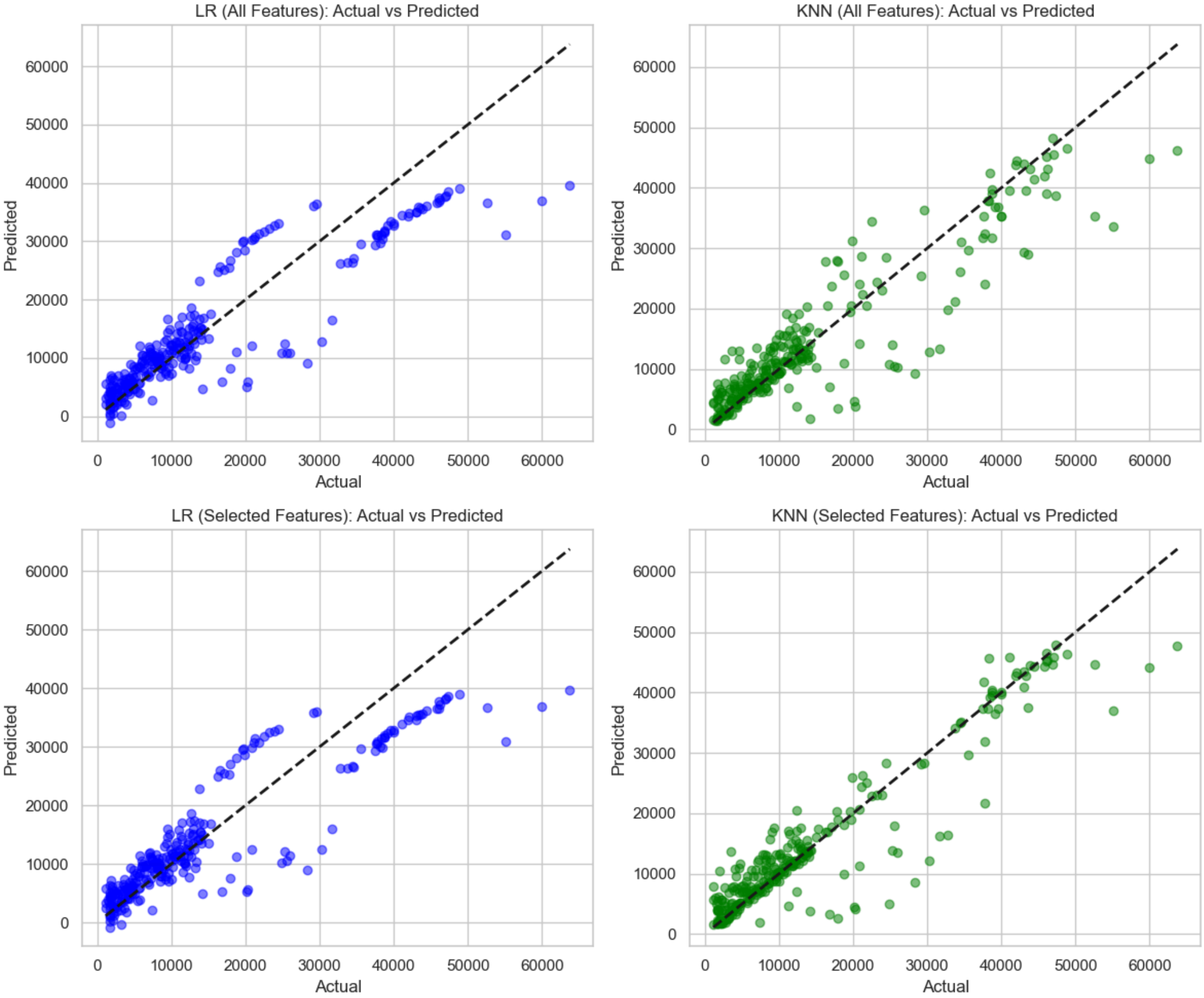
- Correlation matrices showing relationships between features and target :



- Actual vs Predicted scatter plots for both models



- Performance comparison charts



LR (All Features): Actual vs Predicted

KNN (All Features): Actual vs Predicted

LR (Selected Features): Actual vs Predicted

KNN (Selected Features): Actual vs Predicted

Part 2: Fashion-MNIST Image Classification

General Information on Dataset

- **Name**: Fashion-MNIST Dataset
- **Classes**: 5 classes (T-shirt/top, Trouser, Pullover, Dress, Coat)
- **Labels**:
  - 0: T-shirt/top
  - 1: Trouser
  - 2: Pullover
  - 3: Dress
  - 4: Coat
- **Total Samples**: 45,000 samples used in this analysis (subset of full dataset)
- **Size of Each Image**: 28x28 grayscale pixels (784 features per image)
- **Training Samples**: 24,000
- **Validation Samples**: 6,000
- **Testing Samples**: 5,000

Implementation Details

Feature Extraction Phase

- **Features Extracted**: Initially 784 features (28x28 pixels)

- **Dimension Reduction**: Applied PCA to reduce from 784 to 100 dimensions
- **Post-PCA Features**: 100 features representing principal components
- **Variance Preserved**: ~93.6% of original variance maintained

Cross-Validation

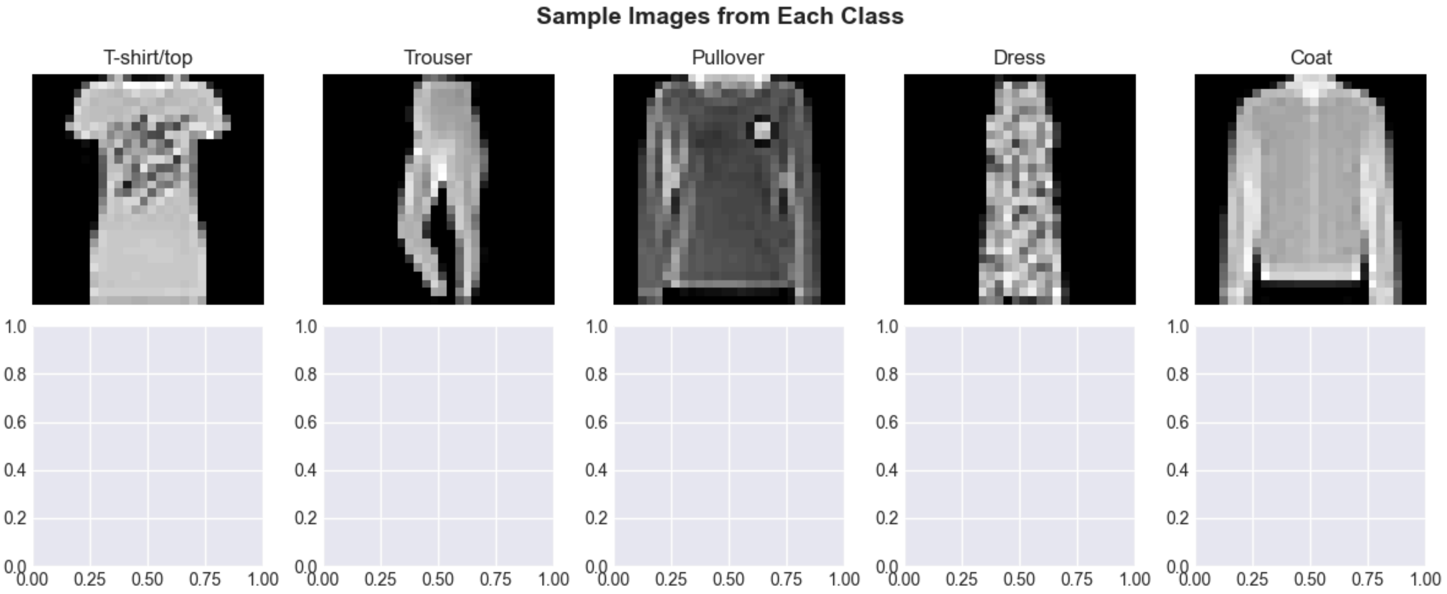
- **Used**: Yes, 5-fold cross-validation
- **Purpose**: Hyperparameter tuning for Logistic Regression
- **Training/Validation Ratio**: 80/20 split within cross-validation folds
- **Grid Search**: Performed with parameters C=[0.1, 1, 10] and solvers=['lbfgs', 'saga']

Hyperparameters Used

Logistic Regression:

- **Solver**: 'lbfgs' (selected via grid search)
- **Regularization Parameter (C)**: 1 (selected via grid search)
- **Max Iterations**: 1000
- **Multi-class Strategy**: 'multinomial' (for multi-class classification)
- **Random State**: 42 (for reproducibility)

  K-Means Clustering:
- **Number of Clusters (k)**: 5 (to match the number of classes)
- **Initialization**: 'k-means++' (for better centroid initialization)
- **Max Iterations**: 300
- **Number of Runs**: 10 (with best result retained)
- **Random State**: 42 (for reproducibility)



Results Details

Logistic Regression Results

- **Test Accuracy** : 88.48%
- **Best Hyperparameters**: C=1, solver='lbfgs'
- **Cross-Validation Score**: ~88.59%
- **Per-Class Performance**:
  - T-shirt/top: 91% precision, 91% recall
  - Trouser: 98% precision, 96% recall
  - Pullover: 84% precision, 80% recall

- Dress: 86% precision, 88% recall
- Coat: 83% precision, 87% recall

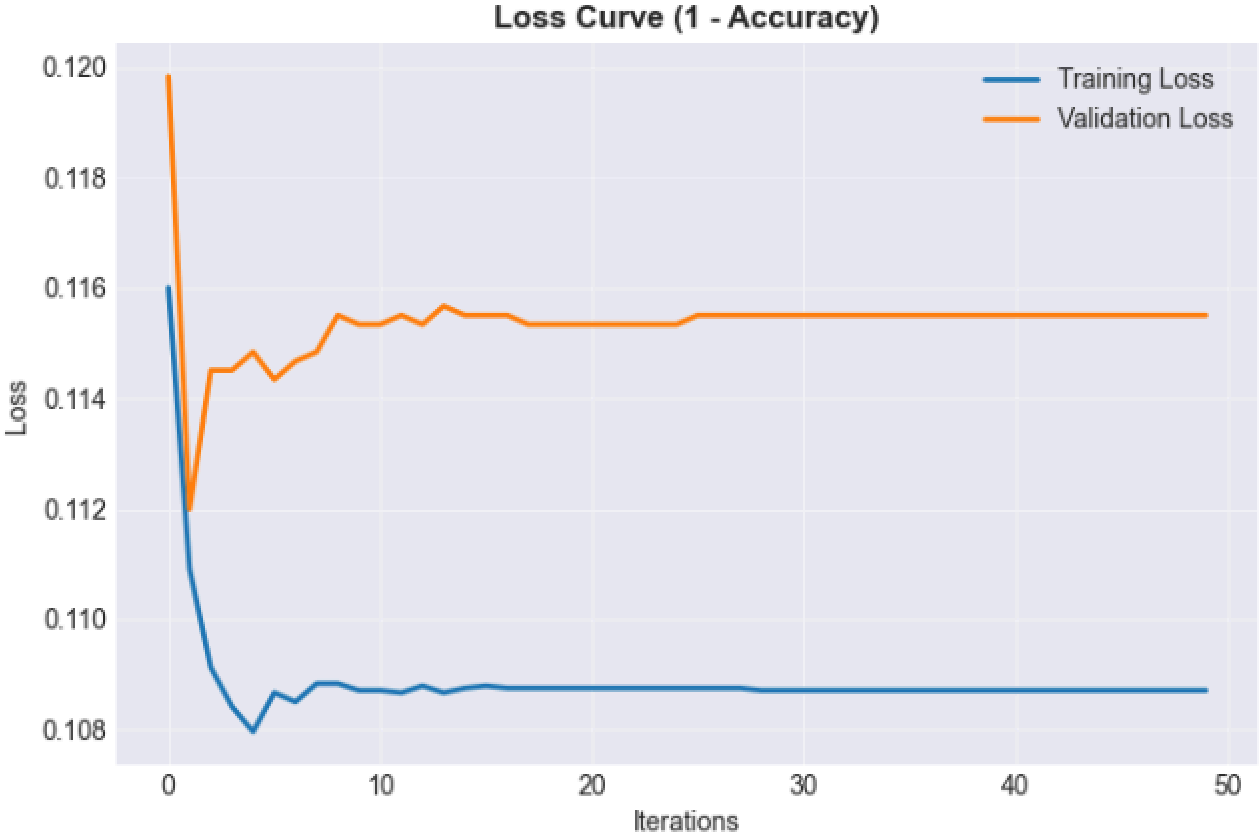
K-Means Clustering Results

- **Test Accuracy**: 55.72%
- **Per-Class Performance**:
  - T-shirt/top: 96% precision, 51% recall
  - Trouser: 73% precision, 86% recall
  - Pullover: 28% precision, 32% recall
  - Dress: 55% precision, 45% recall
  - Coat: 48% precision, 66% recall

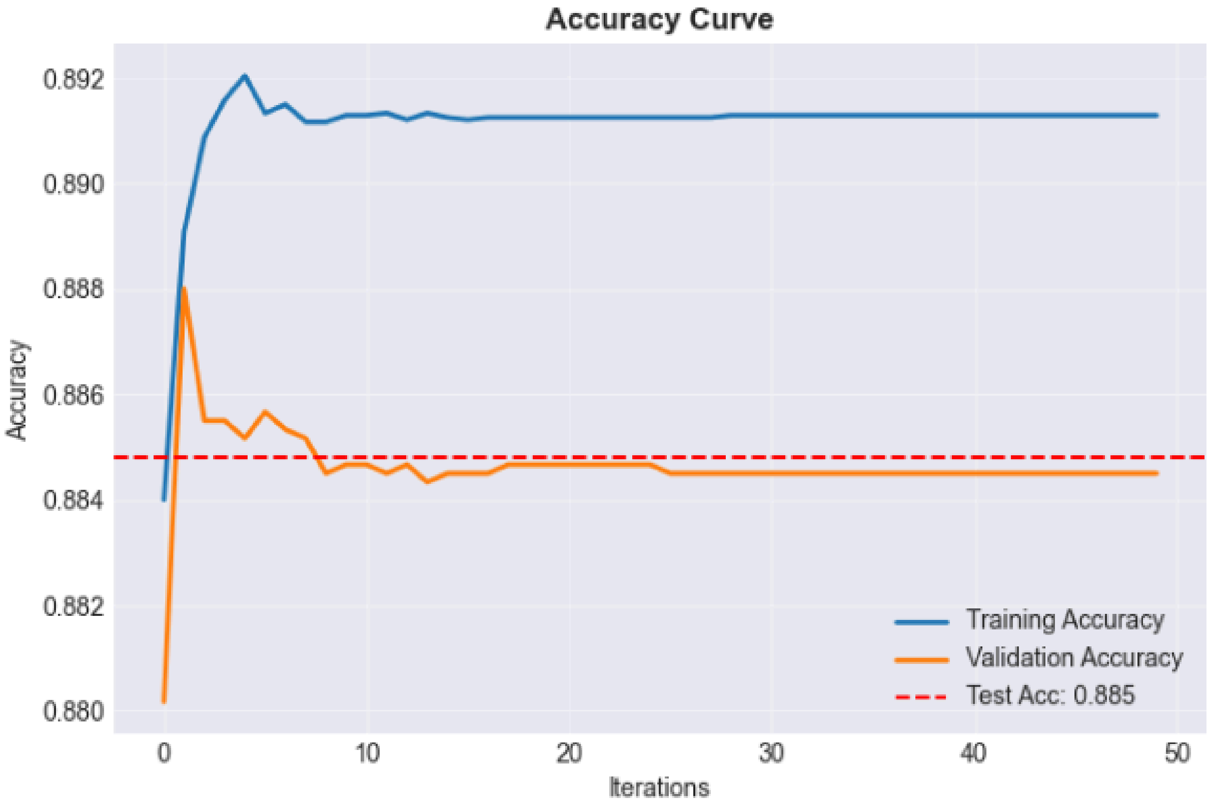
Visualization Results

Logistic Regression:

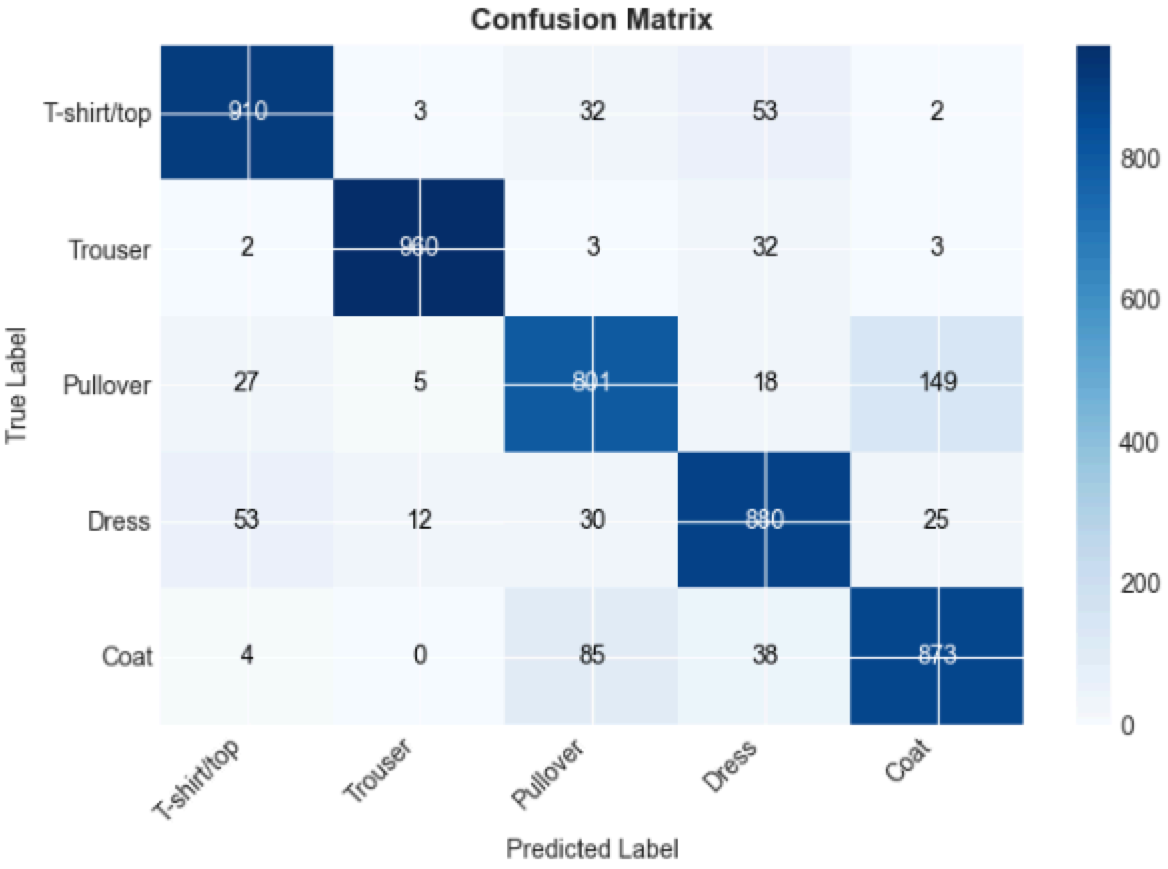
- Loss curve (training/validation progression)



- Accuracy curve showing convergence

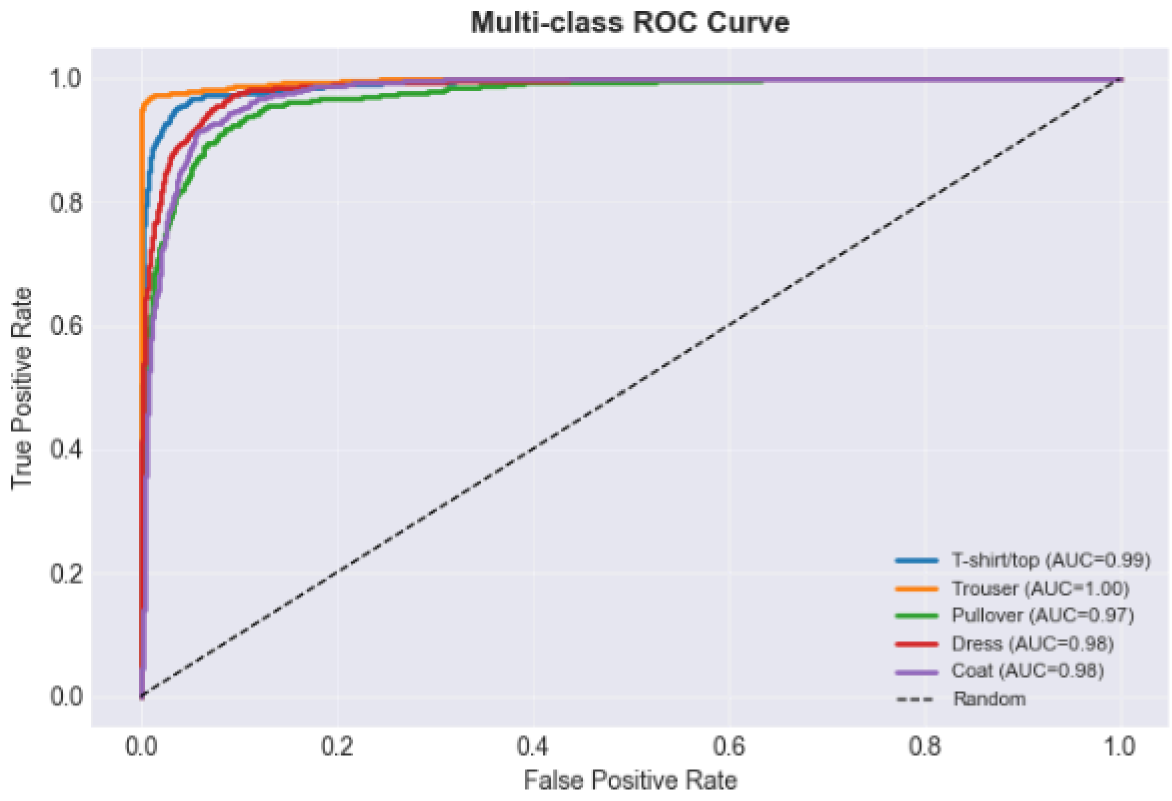


- Confusion matrix showing true vs predicted labels



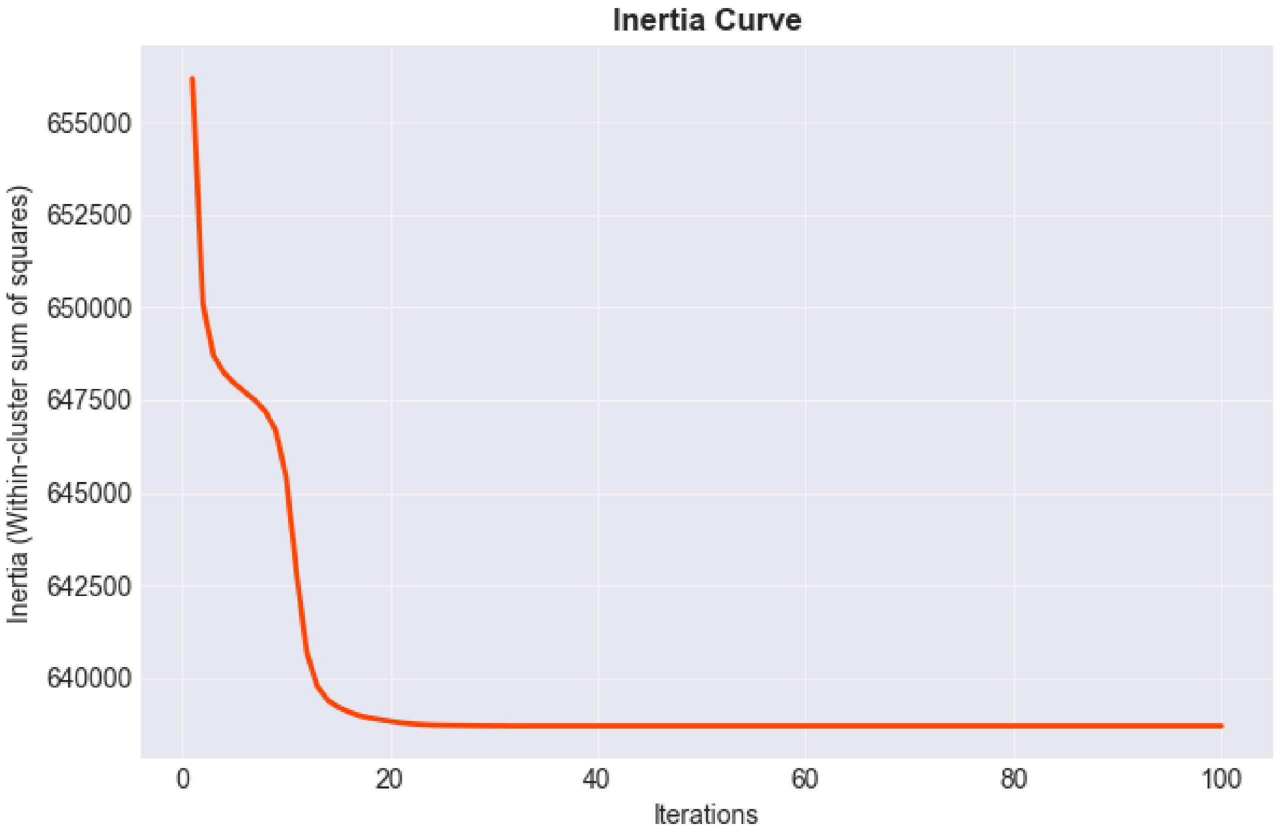
- Multi-class ROC curves for each class

Iterations

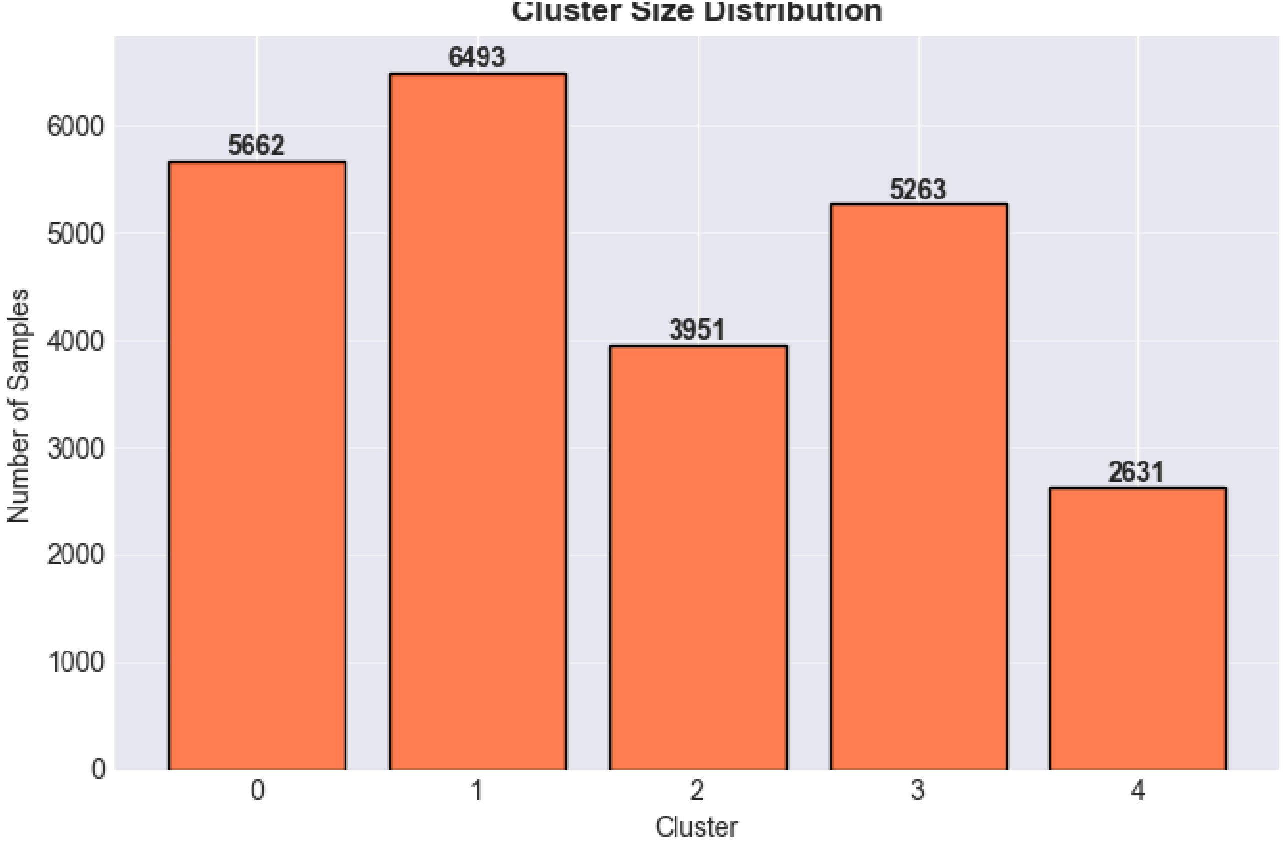


- **K-Means Clustering:**

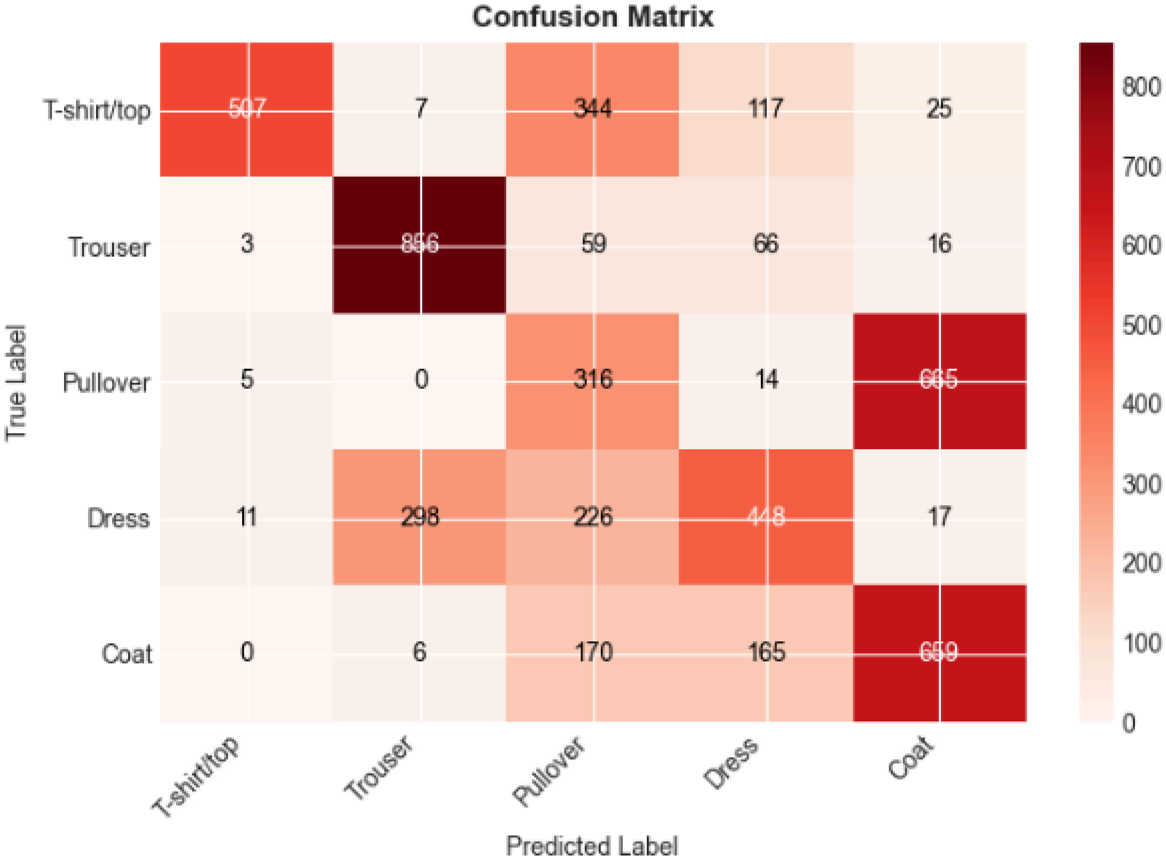
- Inertia curve showing within-cluster sum of squares



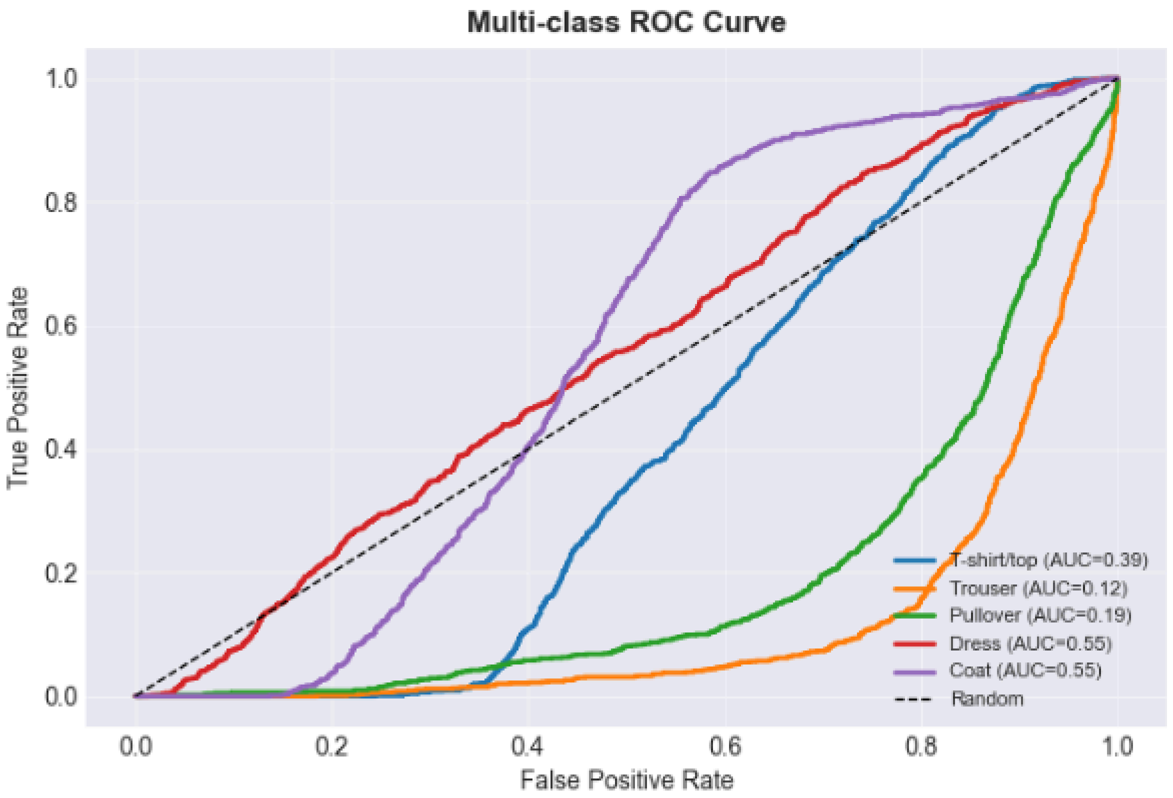
- Cluster size distribution showing number of samples per cluster



- Confusion matrix showing clustering performance



- Multi-class ROC curves based on distance-based probabilities



Model Comparison Summary

Model	Test Accuracy	Approach	Type
Logistic Regression	88.48%	Supervised	Classification
K-Means	55.72%	Unsupervised	Clustering (then classification)

Key Observations

1. **Supervised vs Unsupervised**: Logistic Regression (supervised) significantly outperformed K-Means (unsupervised) by ~32.76%
2. **Feature Learning**: Logistic Regression learns discriminative boundaries between classes
3. **Pattern Recognition**: K-Means learns representative prototypes but struggles with class alignment
4. **Label Importance**: The significant performance gap highlights the importance of labeled data in classification

Technical Implementation Notes

Preprocessing Pipeline

- Data normalized from 0-255 to 0-1 range
- Categories converted to dummy variables for regression analysis
- PCA applied for dimensionality reduction in classification analysis
- Train/validation/test splits performed appropriately for each task

Model Evaluation

- Multiple metrics computed (accuracy, precision, recall, F1-score)
- Confusion matrices generated for detailed performance analysis
- ROC curves plotted for multi-class classification assessment

- Cross-validation used for robust hyperparameter selection