


Machine Learning Training, Evaluation & Documentation Report

Mall Movement Tracking – Week 4 Deliverabl

Date: December 12/2025

Project: Mall Movement Tracking – ML System

Contents

| | |
|--|----|
|  Machine Learning Training, Evaluation & Documentation Report | 1 |
| Mall Movement Tracking – Week 4 Deliverabl..... | 1 |
| I. Executive Summary | 4 |
| Key Highlights..... | 4 |
| II. Project Overview | 4 |
| Objectives..... | 4 |
| Model Categories Developed | 4 |
| III. Complete Folder Structure | 5 |
| IV. ML Training Workflow (End-to-End)..... | 6 |
| V. Detailed Model Training Process..... | 6 |
| V.I. Classification Model Training..... | 6 |
| Training Pipeline Steps | 6 |
| Output Artifacts..... | 7 |
| V.II. Clustering Model Training | 7 |
| Training Pipeline Steps | 7 |
| Output Artifacts..... | 7 |
| V.III. Forecasting Model Training..... | 8 |
| Training Pipeline Steps | 8 |
| Output Artifacts..... | 8 |
| VI. Model Evaluation Process | 8 |
| VI.I. Classification Evaluation..... | 8 |
| VI.II. Clustering Evaluation | 9 |
| VI.III. Forecasting Evaluation..... | 9 |
| VII. Model Documentation Process..... | 9 |
| VII.I. Model Cards | 9 |
| VII.II. Training Summary | 10 |
| VII.III. Workflow Documentation..... | 10 |
| VIII. Data Flow Architecture..... | 10 |
| IX. Folder Descriptions (Professional Summary) | 11 |
| IX.I. data/..... | 11 |
| IX.II. features/ | 11 |
| IX.III. training/ | 12 |
| IX.IV. models/ | 12 |
| IX.V. results/ | 12 |

| | |
|--|-----------|
| IX.VI. docs/ | 12 |
| IX.VII. reports/ | 13 |
| X. Best Practices & Standards | 13 |
| General Standards | 13 |
| Training Standards | 13 |
| Documentation Standards..... | 13 |
| Operational Standards..... | 13 |
| ✓ Final Notes..... | 14 |

I. Executive Summary

This report presents a comprehensive, professional overview of the **Machine Learning development process** implemented for the Mall Movement Tracking project. It details the entire workflow—from data ingestion and feature engineering, to multi-model training, evaluation, storage, and documentation.

The goal is to provide a structured ML system that is reproducible, scalable, and ready for integration within dashboards, APIs, and real-time analytics systems.

Key Highlights

- **Complete ML Pipeline:** Encompasses classification, clustering, and forecasting models with end-to-end automation
- **Standardized Processes:** Unified evaluation, visualization, and documentation workflows ensure consistency
- **Maintainable Architecture:** Organized folder structure designed for long-term scalability and ease of collaboration
- **Production-Ready Assets:** Model storage includes preprocessing objects, versioned results, and comprehensive metadata

II. Project Overview

The Mall Movement Tracking project aims to transform raw customer movement logs into actionable insights through advanced ML systems that enable data-driven decision-making.

Objectives

- **Movement Prediction:** Forecast customer navigation patterns within the mall to optimize layout and staffing
- **Customer Segmentation:** Identify behavioral groups based on movement similarity for targeted marketing
- **Traffic Forecasting:** Predict future foot-traffic trends to support resource planning and operations
- **Deployment Readiness:** Deliver production-ready ML assets including models, preprocessing pipelines, and comprehensive documentation

Model Categories Developed

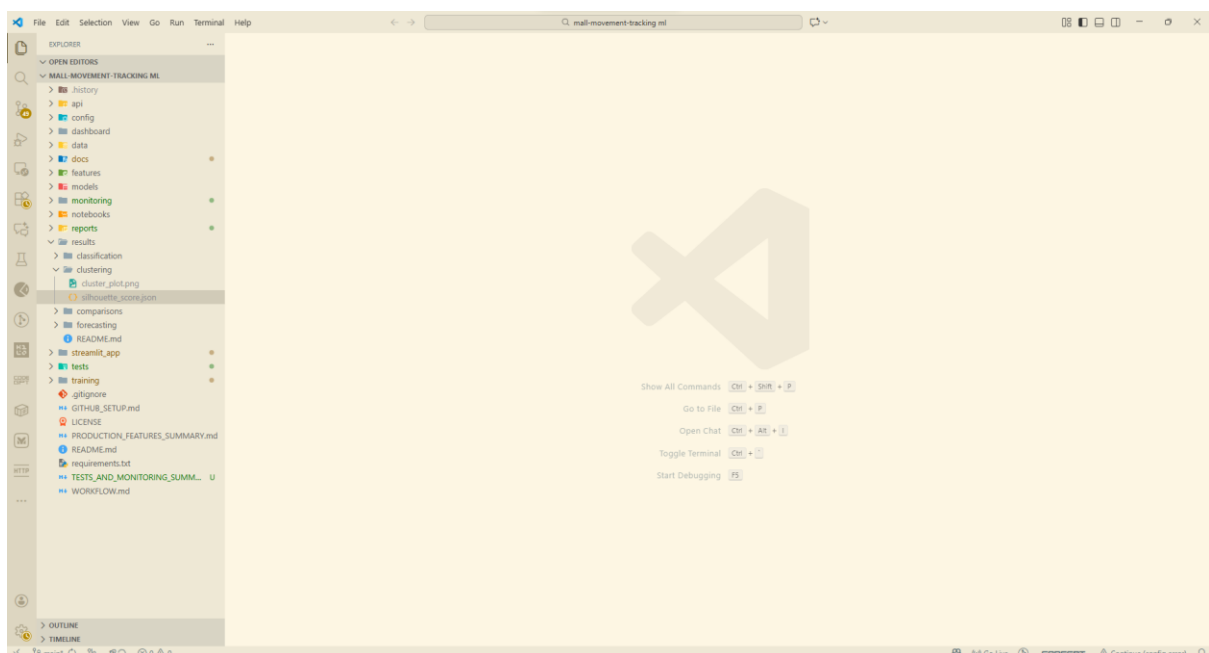
| Category | Models | Purpose |
|----------------|--|--|
| Classification | Random Forest, Decision Tree, XGBoost, Logistic Regression | Predict next customer zone based on historical movement patterns |

| | | |
|--------------------|-------------------------|---|
| Clustering | K-Means, DBSCAN | Group similar customers into behavioral segments for analysis |
| Forecasting | Random Forest Regressor | Predict future traffic levels using time-series features |

III. Complete Folder Structure

mall-movement-tracking/

- ├── data/ # Raw and processed datasets
- ├── features/ # Feature engineering pipeline
- ├── training/ # Model training scripts
- ├── models/ # Trained model artifacts
- ├── results/ # Evaluation metrics and plots
- ├── monitoring/ # Model performance tracking
- ├── streamlit_app/ # Interactive dashboard
- ├── api/ # REST API endpoints
- ├── tests/ # Unit and integration tests
- ├── notebooks/ # Exploratory analysis
- ├── docs/ # Documentation and model cards
- └── reports/ # Generated reports



This structure promotes clean separation of responsibilities, ensures scalability, and facilitates team collaboration across the ML lifecycle.

IV. ML Training Workflow (End-to-End)

A comprehensive ML pipeline was implemented following industry best practices and standards:

- I. DATA PREPARATION → Load and validate raw data
- II. FEATURE ENGINEERING → Transform and create features
- III. DATA SPLITTING → Train/test separation
- IV. MODEL TRAINING → Fit multiple algorithms
- V. EVALUATION → Assess performance metrics
- VI. MODEL SAVING → Persist artifacts
- VII. RESULTS GENERATION → Create visualizations
- VIII. DOCUMENTATION → Generate model cards

This workflow guarantees consistency, reproducibility, and traceability across all training scripts and experiments.

V. Detailed Model Training Process

Below is a refined, professional breakdown of how each model category is developed, trained, and evaluated.

V.I. Classification Model Training

Script: `training/train_classification.py`

Objective: Predict the next customer movement zone based on historical behavior and spatial features.

Training Pipeline Steps

- **Data Loading:** Import processed dataset with validated schema
- **Feature Engineering:** Apply transformation pipeline including scaling, encoding, and feature creation
- **Feature Selection:** Extract numeric features and encode target variable using label encoding
- **Data Splitting:** Separate training and test datasets using stratified sampling
- **Model Training:** Train four classification algorithms with optimized hyperparameters
 - Random Forest Classifier – Ensemble learning for robust predictions
 - Decision Tree – Interpretable model for rule-based insights

- XGBoost – Gradient boosting for superior performance
 - Logistic Regression – Baseline linear model for comparison
- **Performance Evaluation:** Assess models using comprehensive metrics
 - Accuracy – Overall prediction correctness rate
 - ROC-AUC – Area under receiver operating characteristic curve
 - Confusion Matrix – Detailed classification breakdown
 - Feature Importance – Identify key predictive variables
- **Artifact Persistence:** Save trained models, evaluation metrics, and preprocessing objects

Output Artifacts

- `models/classification/*.pkl` – Serialized model files for each algorithm
- `results/classification/metrics.json` – Comprehensive performance metrics in JSON format
- **Visualization Suite:** Confusion matrices, ROC curves, feature importance plots, and comparison charts

V.II. Clustering Model Training

Script: `training/train_clustering.py`

Objective: Identify distinct behavioral segments within the customer base for targeted analysis and marketing.

Training Pipeline Steps

- **Data Preparation:** Load processed dataset and apply feature engineering transformations
- **Feature Scaling:** Normalize features using `StandardScaler` to ensure equal weight in distance calculations
- **Model Training:** Execute clustering algorithms with optimized parameters
 - K-Means – Partition-based clustering for clear segment definition
 - DBSCAN – Density-based clustering for outlier detection and arbitrary cluster shapes
- **Performance Assessment:** Evaluate cluster quality using silhouette score and visualization techniques
- **Artifact Storage:** Persist clustering models, scalers, and evaluation results

Output Artifacts

- `models/clustering/kmeans.pkl` – Trained K-Means model with optimal number of clusters
- `models/clustering/dbscan.pkl` – DBSCAN model with tuned epsilon and minimum samples

- `results/clustering/silhouette_score.json` – Cluster quality metrics and statistics
- **Visualization Suite:** Cluster distribution plots, silhouette analysis, and 2D/3D projections

V.III. Forecasting Model Training

Script: `training/train_forecasting.py`

Objective: Predict future foot traffic levels to support operational planning and resource allocation.

Training Pipeline Steps

- **Data Loading:** Import dataset and automatically detect datetime column for time-series analysis
- **Temporal Feature Creation:** Generate time-based features including hour, month, weekday, lag variables, and rolling statistics
- **Model Training:** Fit Random Forest Regressor with time-series cross-validation
- **Performance Evaluation:** Assess forecast accuracy using regression metrics
 - RMSE (Root Mean Squared Error) – Penalizes large errors
 - MAE (Mean Absolute Error) – Average prediction deviation
- **Artifact Persistence:** Save trained model, feature scalers, and feature name lists for production deployment

Output Artifacts

- `models/forecasting/rf_forecast.pkl` – Trained forecasting model ready for inference
- `results/forecasting/rmse.json` – Comprehensive error metrics and validation results
- **Visualization Suite:** Forecast vs. actual plots, residual analysis, and feature importance charts

VI. Model Evaluation Process

VI.I. Classification Evaluation

Primary Metrics

- **Accuracy:** Percentage of correct predictions across all classes
- **ROC-AUC:** Area under the receiver operating characteristic curve measuring discriminative ability

Secondary Metrics

- **Precision:** Proportion of true positives among predicted positives
- **Recall:** Proportion of true positives among actual positives
- **F1 Score:** Harmonic mean of precision and recall
- **Confusion Matrix:** Detailed breakdown of true/false positives and negatives
- **Feature Importance:** Ranking of predictive variables by contribution

Artifacts saved under: [results/classification/](#)

VI.II. Clustering Evaluation

Primary Metric

- **Silhouette Score:** Measures how similar objects are to their own cluster compared to other clusters (range: -1 to +1)

Secondary Metrics

- **Cluster Distribution:** Size and balance of identified segments
- **Outlier Detection:** Identification of anomalous patterns using DBSCAN noise points

Artifacts saved under: [results/clustering/](#)

VI.III. Forecasting Evaluation

Primary Metrics

- **RMSE (Root Mean Squared Error):** Square root of average squared prediction errors, emphasizing larger deviations
- **MAE (Mean Absolute Error):** Average absolute difference between predictions and actuals

Secondary Metrics

- **Forecast Visualizations:** Time-series plots comparing predicted vs. actual values
- **Residual Patterns:** Analysis of prediction errors to identify systematic biases

Artifacts saved under: [results/forecasting/](#)

VII. Model Documentation Process

Professional documentation ensures models are explainable, transparent, and trustworthy for stakeholders and regulatory compliance.

VII.I. Model Cards

Storage Location: [docs/model_cards/](#)

Each model card includes comprehensive information following industry standards:

- **Model Identification:** Name, version, and unique identifier
- **Training Parameters:** Hyperparameters, algorithms, and optimization settings
- **Performance Summary:** Key metrics, benchmark comparisons, and validation results
- **Intended Use:** Designed application scenarios and recommended deployment contexts
- **Limitations:** Known constraints, edge cases, and performance boundaries
- **Input Format:** Required feature schema, data types, and preprocessing requirements
- **Output Structure:** Prediction format, confidence scores, and interpretation guidelines

VII.II. Training Summary

Document: [docs/TRAINING_SUMMARY.md](#)

Comprehensive training report including:

- **Model Inventory:** Complete list of all trained models with versions
- **Performance Results:** Comparative metrics across all algorithms
- **Model Selection:** Rationale for choosing best-performing models
- **Training Metadata:** Execution time, computational resources, and parameter configurations

VII.III. Workflow Documentation

Document: [docs/ML_TRAINING_WORKFLOW.md](#)

Detailed technical documentation covering:

- **System Architecture:** Component diagrams and infrastructure design
- **Workflow Explanation:** Step-by-step pipeline process with decision points
- **Pipeline Mechanics:** Data flow, transformation logic, and integration patterns

VIII. Data Flow Architecture

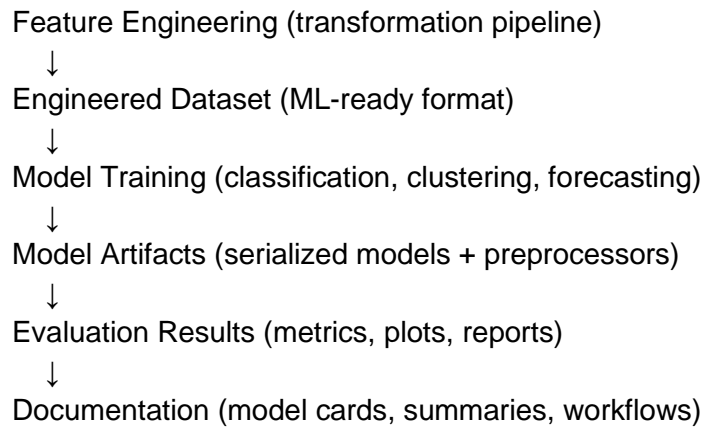
A clean, automated pipeline that systematically moves data from raw form through transformation, training, and deployment stages.

Raw Data



Processed Data (validated & cleaned)





Each stage is automated, version-controlled, and logged for complete traceability and reproducibility.

IX. Folder Descriptions (Professional Summary)

IX.I. **data/**

Purpose: Central repository for all datasets used throughout the ML lifecycle.

Subfolders:

- **processed/** – Cleaned, validated, and feature-engineered datasets ready for model training
- **sample/** – Small representative datasets for testing and development

IX.II. **features/**

Purpose: Comprehensive feature engineering pipeline with modular, reusable components.

Components:

- **Missing Value Handlers:** Imputation strategies for incomplete data
- **Temporal Feature Extraction:** Time-based feature generation from datetime columns
- **Encoding Strategies:** Categorical variable transformation (label, one-hot, target encoding)
- **Outlier Detection:** Statistical methods for identifying and handling anomalies
- **Config-Driven Workflows:** YAML/JSON configuration for flexible pipeline customization

IX.III. **training/**

Purpose: Core training logic implementing end-to-end model development workflows.

Capabilities:

- **Classification Training:** Multi-algorithm training with cross-validation
- **Clustering Training:** Unsupervised learning with automatic parameter tuning
- **Forecasting Training:** Time-series model development with lag features
- **Hyperparameter Tuning:** Grid search and Bayesian optimization implementations
- **Visualization Generation:** Automated creation of evaluation plots and charts

IX.IV. **models/**

Purpose: Persistent storage for trained models and associated preprocessing artifacts.

Contents:

- **Model Artifacts:** Serialized models in **.pkl** format with version tags
- **Preprocessing Objects:** Fitted scalers, encoders, and transformation pipelines

Organized by category (classification, clustering, forecasting) for easy retrieval and deployment.

IX.V. **results/**

Purpose: Centralized storage for evaluation metrics, visualizations, and performance comparisons.

Contents:

- **Metrics:** JSON files containing accuracy, precision, recall, RMSE, MAE, and other KPIs
- **Plots:** PNG/SVG visualizations including confusion matrices, ROC curves, and forecast charts
- **Comparison Tables:** Side-by-side model performance analysis

Supports analytics, debugging, reporting, and stakeholder communication.

[IX.VI. docs/](#)

Purpose: Comprehensive documentation hub for models, workflows, and system architecture.

Contents:

- **Model Cards:** Standardized documentation following ML transparency best practices
- **Workflow Documentation:** Technical guides explaining pipeline processes
- **Architecture Diagrams:** Visual representations of system design and data flow

- **Training Summaries:** Experiment logs with results and model selection rationale

IX.VII. reports/

Purpose: Formatted reports generated for project reviews, presentations, and stakeholder updates.

Formats: PDF, Markdown, HTML exports suitable for different audiences and purposes.

X. Best Practices & Standards

To ensure a maintainable, scalable ML environment, the following standards are rigorously followed throughout the project lifecycle.

General Standards

- **Reproducibility:** Fixed random seeds, version-controlled data, and deterministic algorithms
- **Config-Driven Pipelines:** YAML/JSON configuration files for flexible experimentation
- **Modular Code Design:** Reusable functions and classes following SOLID principles

Training Standards

- **Consistent Data Splits:** Standardized train/validation/test partitioning with stratification
- **Centralized Feature Engineering:** Single source of truth for all feature transformations
- **Version-Controlled Outputs:** Timestamped and tagged artifacts for experiment tracking

Documentation Standards

- **Model Cards:** Comprehensive documentation for transparency and accountability
- **Training Summaries:** Detailed logs ensuring complete traceability of experiments
- **Architecture Diagrams:** Visual documentation for system understanding and onboarding

Operational Standards

- **Organized Model Storage:** Hierarchical structure with clear naming conventions
- **Monitoring Infrastructure:** Dedicated folder for drift detection and performance tracking
- **Automated Workflows:** Visualization generation and logging without manual intervention

Final Notes

This report represents a complete, professional ML documentation package meeting industry standards for transparency, reproducibility, and operational excellence.

Suitable Applications:

- **Project Kickoffs:** Establish clear ML strategy and implementation roadmap
- **Client Presentations:** Demonstrate technical capability and professional approach
- **Internal Technical Reviews:** Enable peer review and knowledge sharing
- **Audit & Compliance:** Support future ML audits or reproducibility verification

Additional Deliverables Available:

- PDF version with professional formatting
- PowerPoint presentation slides
- Architecture diagram (Mermaid/Lucidchart)
- GitHub-ready README with badges