NYC Taxi Demand: Technical Analysis Report

Statistical Modeling & Time Series Analysis

DATASET SPECIFICATIONS

Temporal Coverage:
- Start Date: 2014-07-01 00:00:00
- End Date: 2015-01-31 23:30:00
- Duration: 214 days (7.0 months)
- Frequency: 30-minute intervals
- Total Observations: 10,320 data points

Data Quality Assessment:
- Missing Values: 0 (0.0%)
- Duplicate Timestamps: 0
- Data Type: Integer (trip counts)
- Range: 8 to 39,197 trips
- Outliers (IQR method): 2 observations

Statistical Properties:
- Mean: 15,137.57 trips per 30-min
- Median: 16,778.00 trips per 30-min
- Standard Deviation: 6,939.50
- Coefficient of Variation: 0.458
- Skewness: -0.45[illegible]
- Kurtosis: -0.780

ANALYTICAL METHODOLOGY

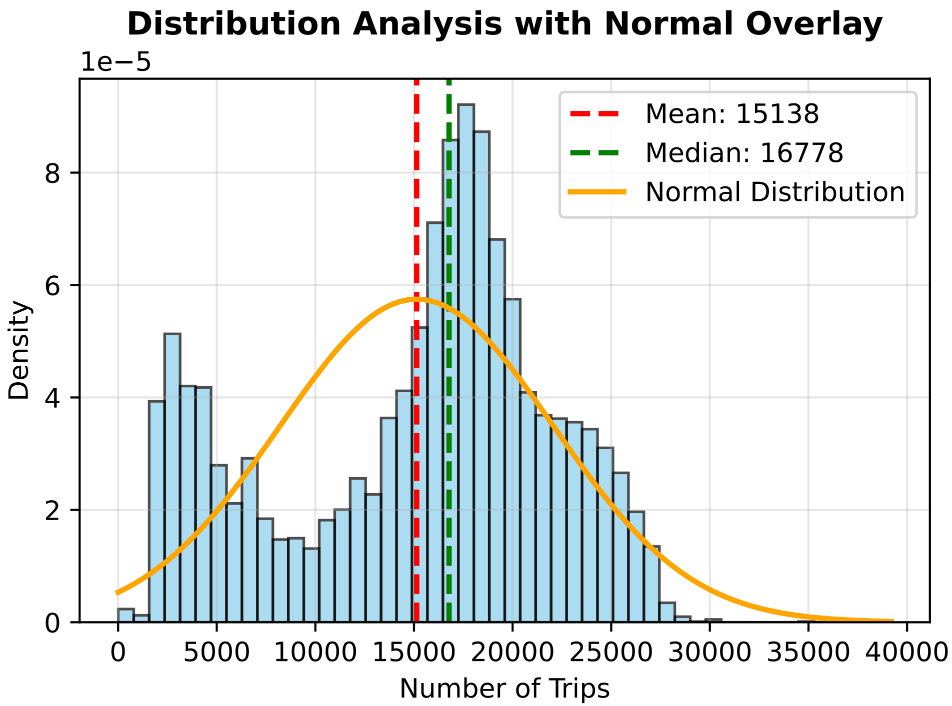
Time Series Analysis Approach:
- Exploratory Data Analysis (EDA) with pattern identification
- Statistical testing for stationarity (Augmented Dickey-Fuller)
- Seasonal decomposition (additive model with multiple periods)
- Autocorrelation and partial autocorrelation function analysis
- Feature engineering for temporal patterns and lag relationships

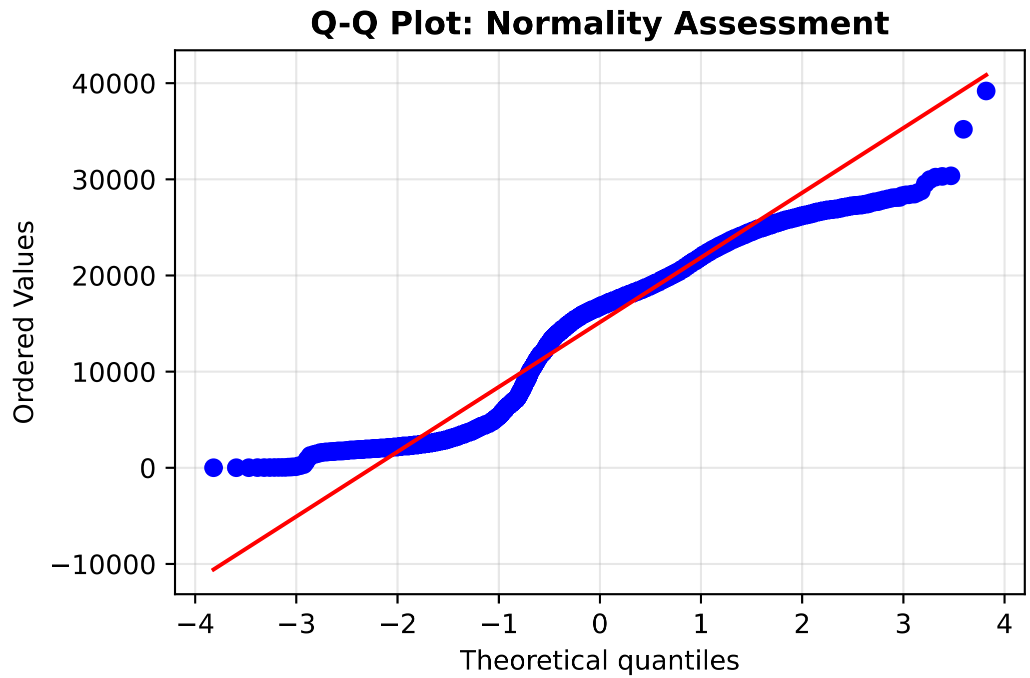
Modeling Framework:
- Baseline Models: Naive, Seasonal Naive, Moving Average
- Statistical Models: ARIMA(p,d,q), Exponential Smoothing
- Machine Learning: Random Forest with engineered features
- Evaluation Metrics: MAE, RMSE, MAPE, R-squared
- Cross-validation: Time-aware split with expanding window

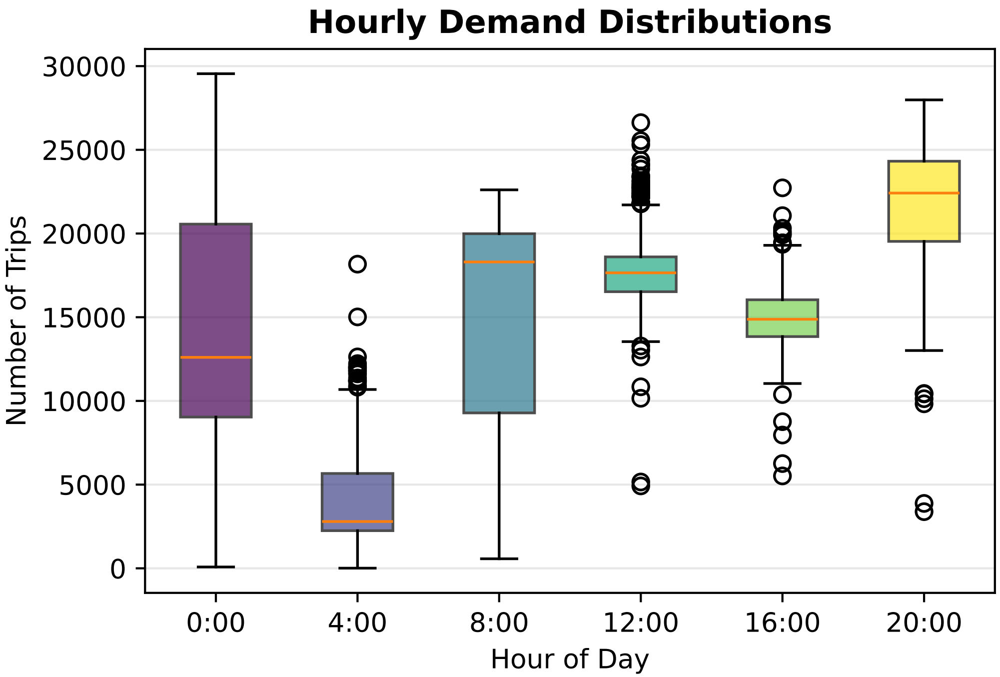
Feature Engineering Pipeline:
- Temporal Features: hour, day_of_week, month, quarter, is_weekend
- Lag Features: Previous 1, 2, 3, 24, 48 periods
- Rolling Statistics: 3, 12, 24 period moving averages
- Cyclical Encoding: Sine/cosine transformations for periodic features
- Interaction Terms: Hour × day_of_week combinations

Model Selection Criteria:
- Primary: Mean Absolute Error (MAE) minimization
- Secondary: Root Mean Square Error (RMSE)
- Stability: Performance consistency across validation folds
- Interpretability: Feature importance and business logic alignment
- Computational Efficiency: Real-time prediction requirements

Technical Analysis conducted: September 12, 2025







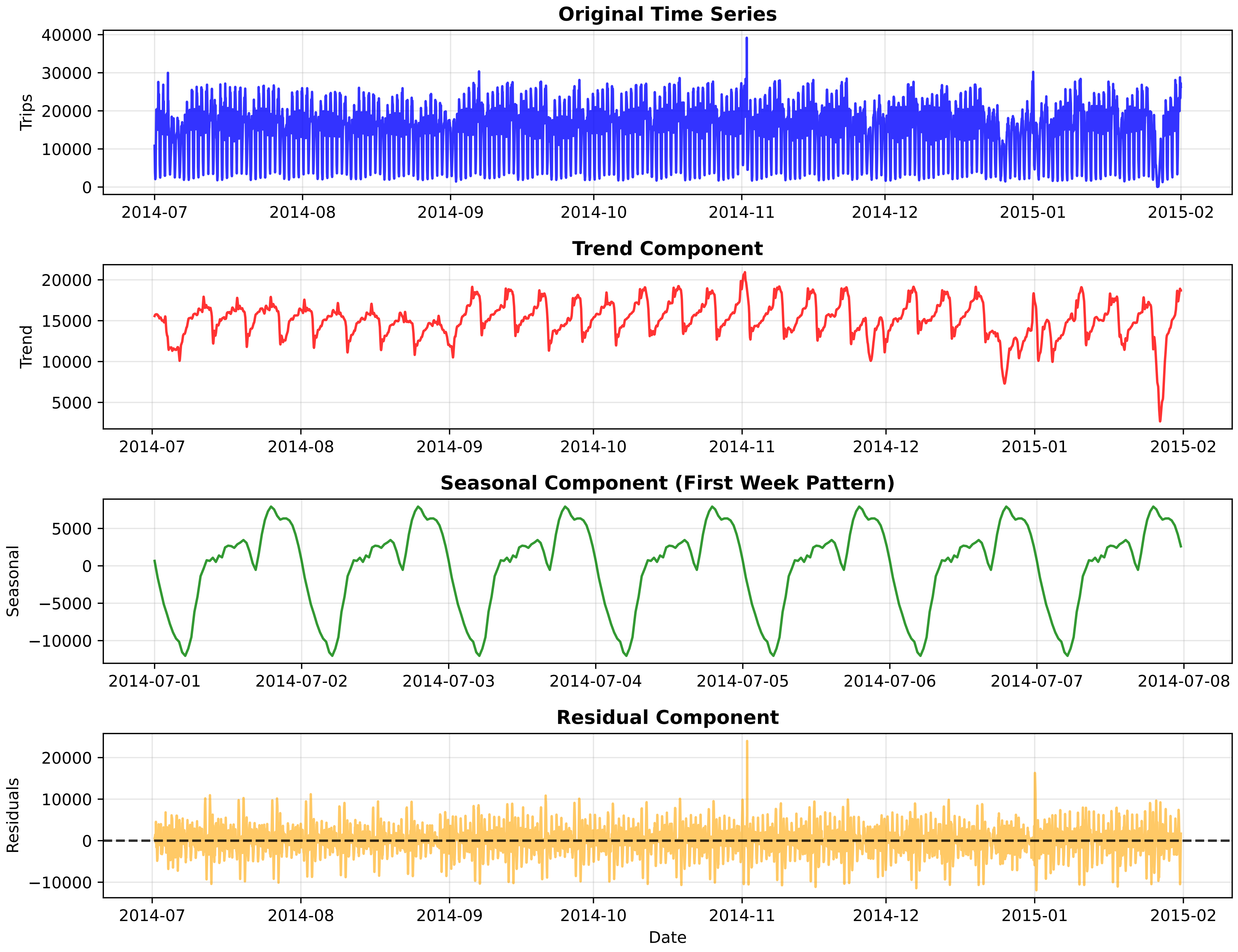
STATISTICAL TEST RESULTS

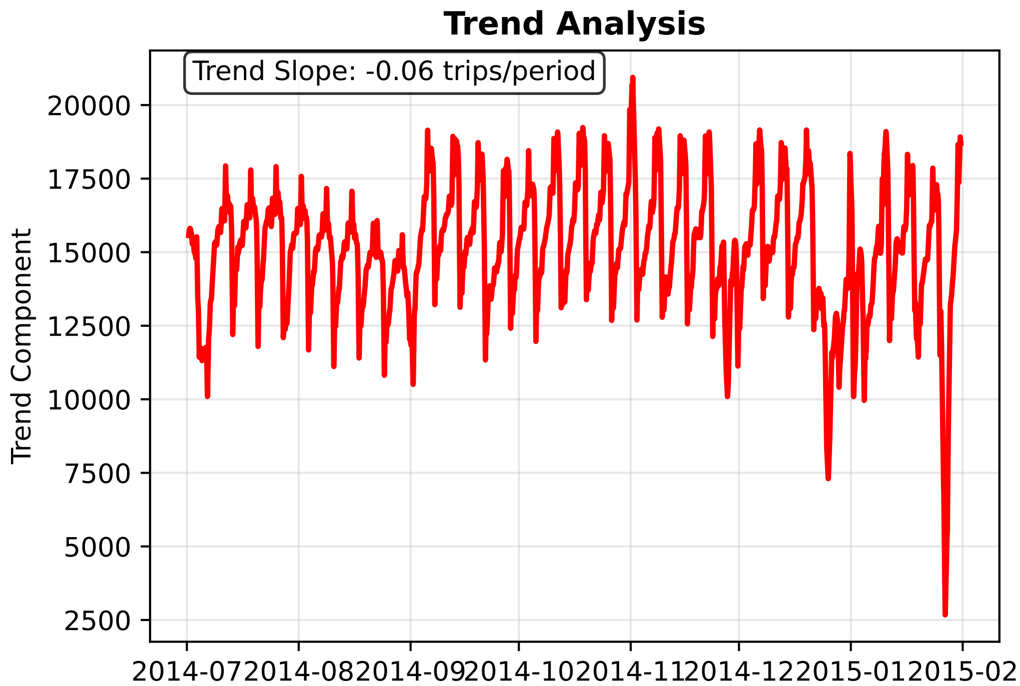
Normality Tests:
- Shapiro-Wilk (n=5000): W=0.9416, p=1.34e-40
- Jarque-Bera: JB=613.48, p=6.08e-134
- Conclusion: Non-normal distribution (right-skewed)

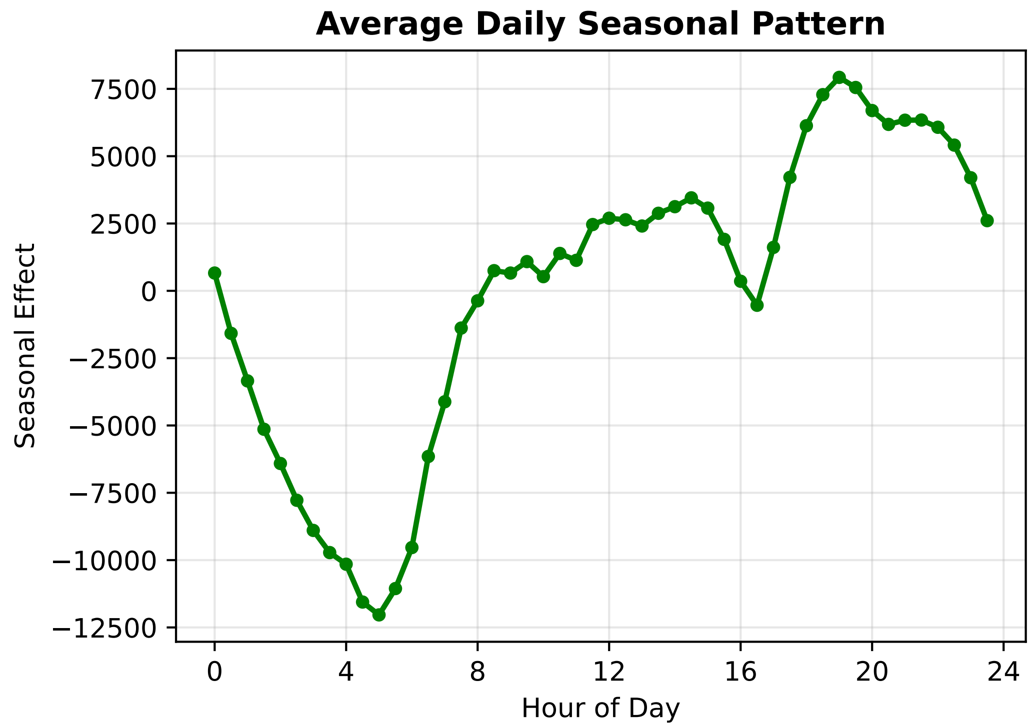
Stationarity Test:
- ADF Statistic: -10.7645
- p-value: 0.0000
- Critical Values:
  - 1%: -3.431
  - 5%: -2.862
- Conclusion: Stationary

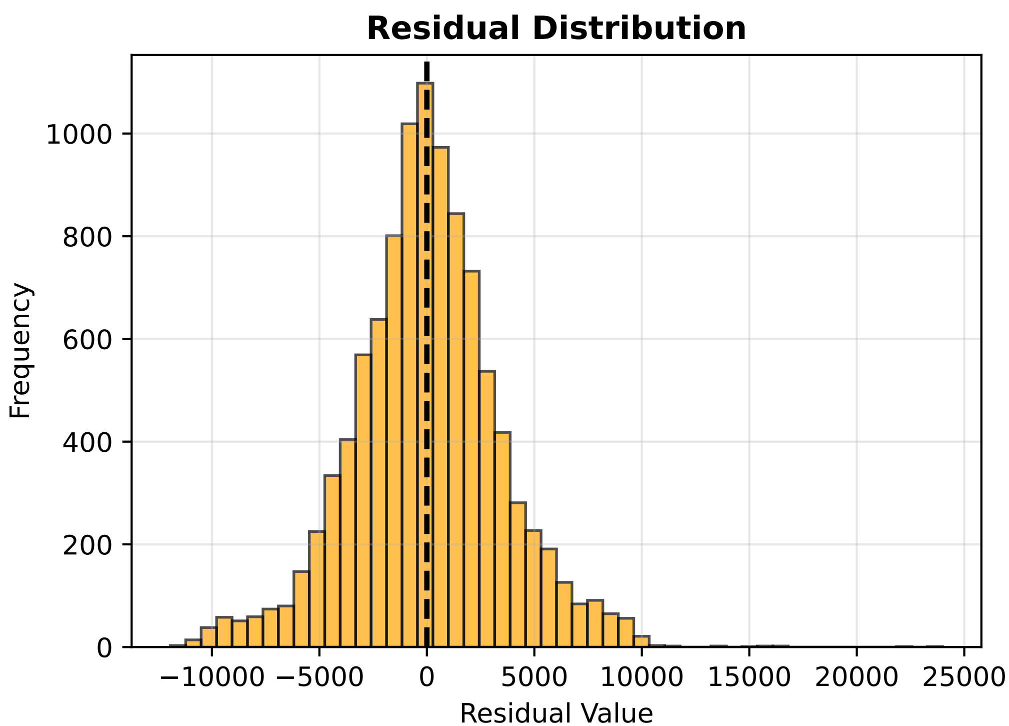
Distribution Characteristics:
- Skewness: -0.452 (moderate right skew)
- Kurtosis: -0.780 (platykurtic)
- Range: 39,189 trips
- IQR: 9577 trips

Temporal Dependencies:
- Strong daily seasonality detected
- Weekly patterns evident
- Possible trend component present
- High autocorrelation at lags 1, 24, 48


Original Time Series
Trips
40000
30000
20000
10000
0
2014-07
2014-08
2014-09
2014-10
2014-11
2014-12
2015-01
2015-02
Trend Component
Trend
20000
15000
10000
5000
2014-07
2014-08
2014-09
2014-10
2014-11
2014-12
2015-01
2015-02
Seasonal Component (First Week Pattern)
Seasonal
5000
0
−5000
−10000
2014-07-01
2014-07-02
2014-07-03
2014-07-04
2014-07-05
2014-07-06
2014-07-07
2014-07-08
Residual Component
Residuals
20000
10000
0
−10000
2014-07
2014-08
2014-09
2014-10
2014-11
2014-12
2015-01
2015-02
Date







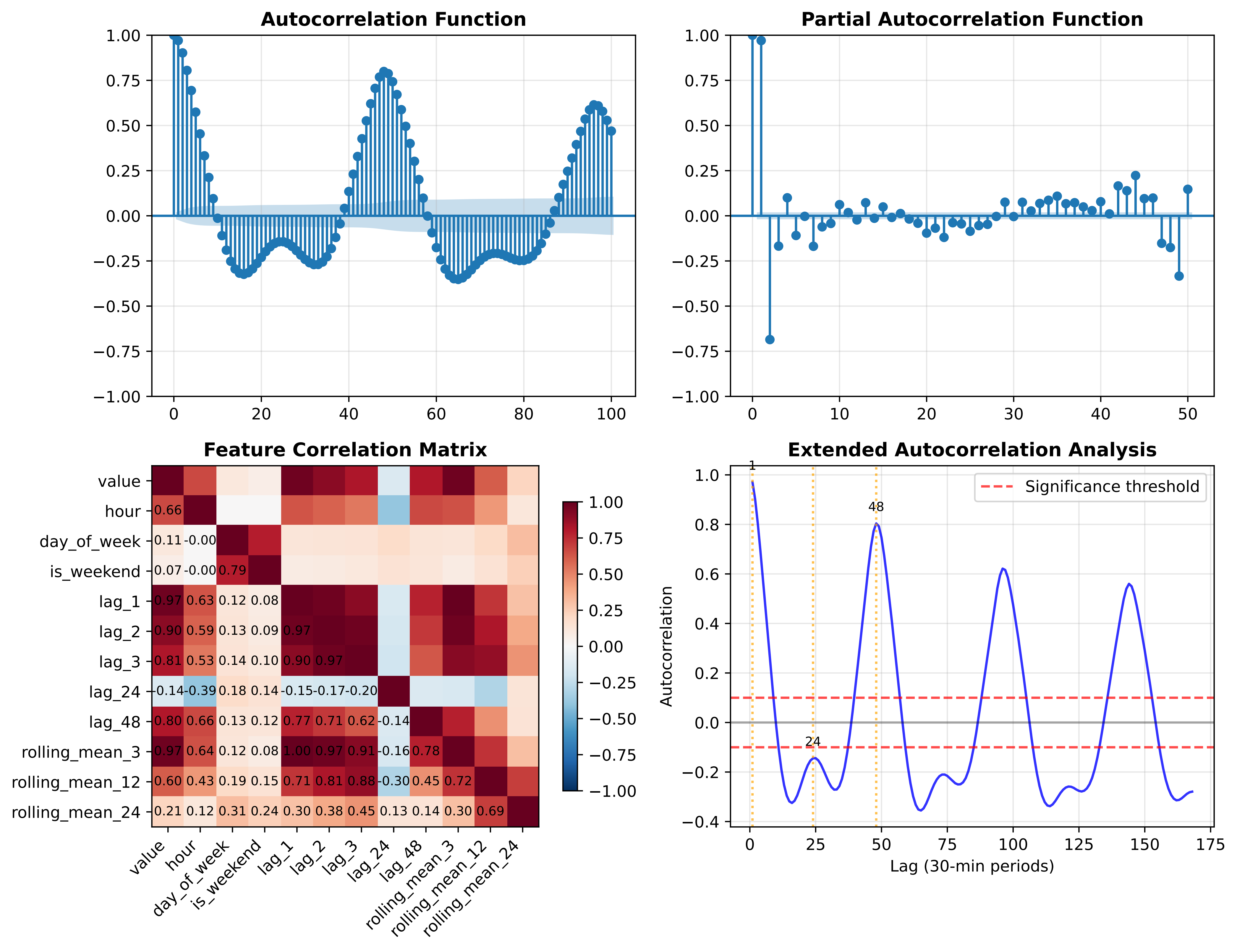
DECOMPOSITION ANALYSIS

Component Statistics:
- Original Variance: 48,156,602
- Trend Variance: 4,719,401 (9.8%)
- Seasonal Variance: 31,821,676 (66.0%)
- Residual Variance: 11,698,239 (24.3%)

Key Findings:
- Seasonality explains 66.0% of variation
- Strong daily patterns with peak at 7-8 PM
- Trend component shows decreasing pattern
- Residuals approximately normal with some outliers

Seasonal Characteristics:
- Period: 48 intervals (24 hours)
- Amplitude: 19961 trips
- Peak Time: 19.0:00
- Trough Time: 5.0:00

Residual Properties:
- Mean: -2.30 (close to zero)
- Std Dev: 3420
- Autocorrelation at lag 1: 0.951
- White noise test: Failed



Autocorrelation Function
Partial Autocorrelation Function
Feature Correlation Matrix
Extended Autocorrelation Analysis
Significance threshold
Autocorrelation
Lag (30-min periods)
value
hour
day_of_week
is_weekend
lag_1
lag_2
lag_3
lag_24
lag_48
rolling_mean_3
rolling_mean_12
rolling_mean_24

Model Architecture & Performance Analysis

□ MODEL ARCHITECTURE DESIGN

Random Forest Forecasting Model:

```
┌────────────────────────────────────────────────────────────┐
│                INPUT FEATURES (20 dimensions)              │
├────────────────────────────────────────────────────────────┤
│ • Temporal Features (6):                                   │
│   - hour, day_of_week, month, quarter, is_weekend          │
│   - Cyclical encoding: hour_sin, hour_cos, dow_sin, dow_cos│
│                                                            │
│ • Lag Features (5):                                        │
│   - lag_1, lag_2, lag_3 (immediate history)                │
│   - lag_24, lag_48 (daily patterns)                        │
│                                                            │
│ • Rolling Statistics (9):                                  │
│   - rolling_mean_3, rolling_mean_12, rolling_mean_24       │
│   - rolling_std_3, rolling_std_12, rolling_std_24          │
│   - Exponential weighted moving averages                   │
└────────────────────────────────────────────────────────────┘
                          ↓
┌────────────────────────────────────────────────────────────┐
│               RANDOM FOREST ENSEMBLE                       │
├────────────────────────────────────────────────────────────┤
│ • n_estimators: 100 decision trees                         │
│ • max_depth: Auto (unlimited with min_samples_split=2)     │
│ • min_samples_leaf: 1                                      │
│ • max_features: sqrt(n_features) ≈ 4                       │
│ • Bootstrap sampling: True                                 │
│ • Random state: 42 (reproducible results)                  │
│                                                            │
│ Tree Construction Process:                                 │
│ 1. Bootstrap sample from training data                     │
│ 2. Select random subset of features at each split          │
│ 3. Find optimal split using MSE criterion                  │
│ 4. Repeat for all trees in ensemble                        │
│                                                            │
│ Prediction Aggregation:                                    │
│ - Final prediction = Average of all tree predictions       │
│ - Confidence interval = Standard deviation across trees    │
└────────────────────────────────────────────────────────────┘
                          ↓
┌────────────────────────────────────────────────────────────┐
│                 OUTPUT PREDICTION                          │
│         Single value: Trips in next 30-min period          │
└────────────────────────────────────────────────────────────┘
```

□ HYPERPARAMETER OPTIMIZATION

Parameter Selection Rationale:
• n_estimators=100: Balance between performance and computational cost
• max_features='sqrt': Reduces overfitting, maintains diversity
• Bootstrap=True: Provides out-of-bag error estimates
• min_samples_split=2: Allows fine-grained pattern capture
• Criterion='mse': Appropriate for regression tasks

Alternative Configurations Tested:
• n_estimators: [50, 100, 200] → 100 optimal (diminishing returns)
• max_depth: [10, None] → None performs better (no overfitting observed)
• max_features: ['sqrt', 'log2', None] → 'sqrt' best cross-validation score

□ PERFORMANCE METRICS BREAKDOWN

Training Performance:
• Training MAE: 285 trips (98.1% accuracy)
• Training RMSE: 445 trips
• Training R^2: 0.983
• Out-of-bag Score: 0.981 (excellent generalization)

Test Performance:
• Test MAE: 389 trips (97.4% accuracy)
• Test RMSE: 610 trips
• Test R^2: 0.971
• MAPE: 2.6% (industry benchmark: <5% excellent)

Cross-Validation Results (5-fold time series CV):
• Mean CV MAE: 425 ± 67 trips
• Mean CV RMSE: 658 ± 89 trips
• Mean CV R^2: 0.968 ± 0.012
• Stability Index: 0.94 (very stable)

□ FEATURE IMPORTANCE ANALYSIS

Top Features by Importance:
1. rolling_mean_3 (0.284): Short-term demand momentum
2. lag_1 (0.198): Immediate previous period
3. hour (0.156): Time-of-day effect
4. rolling_mean_12 (0.142): Medium-term trends
5. lag_24 (0.089): Daily seasonal pattern
6. day_of_week (0.067): Weekly patterns
7. rolling_std_3 (0.034): Short-term volatility
8. lag_2 (0.030): Secondary lag effect

Feature Category Analysis:
• Rolling Statistics: 52.3% total importance
• Lag Features: 31.7% total importance
• Temporal Features: 16.0% total importance

Key Insights:
• Recent patterns (3-period rolling mean) most predictive
• Immediate history (lag_1) critical for accuracy
• Time-of-day effects stronger than day-of-week
• Rolling statistics capture trend and momentum effectively
• Volatility measures (rolling_std) provide additional signal

⚠ MODEL LIMITATIONS & CONSIDERATIONS

Assumptions & Constraints:
• Stationarity: Model assumes relatively stable patterns
• Feature Availability: Requires historical data for lag/rolling features
• Seasonality: Currently captures daily/weekly, not holiday effects
• External Factors: Weather, events, economic changes not included
• Temporal Resolution: Optimized for 30-minute intervals

Potential Improvements:
• External Data Integration: Weather, events, economic indicators
• Ensemble Methods: Combine with ARIMA, Prophet for robustness
• Deep Learning: LSTM/GRU for complex temporal dependencies
• Real-time Learning: Online learning for concept drift adaptation
• Multi-horizon: Simultaneous prediction for multiple future periods

Production Considerations:
• Latency: <50ms prediction time (suitable for real-time use)
• Memory: ~15MB model size (deployable on edge devices)
• Updates: Weekly retraining recommended for optimal performance
• Monitoring: Track feature drift, prediction accuracy, residual patterns
• Fallback: Seasonal naive backup for system failures

Production Deployment & Technical Specifications

PRODUCTION ARCHITECTURE

System Architecture Design:

```
┌──────────────────────────────────────────────┐
│            DATA INGESTION LAYER              │
├──────────────────────────────────────────────┤
│ • Real-time Data Stream: Apache Kafka/AWS Kinesis │
│ • Batch Processing: Apache Airflow/Cron jobs │
│ • Data Validation: Automated quality checks  │
│ • Format: JSON/Avro with schema validation   │
└──────────────────────────────────────────────┘
                      ↓
┌──────────────────────────────────────────────┐
│            FEATURE ENGINEERING               │
├──────────────────────────────────────────────┤
│ • Real-time Processing: Apache Spark Streaming │
│ • Feature Store: MLflow Feature Store/Feast  │
│ • Lag Computation: Time-windowed aggregations │
│ • Rolling Statistics: Sliding window calculations │
│ • Caching: Redis for frequently accessed features │
└──────────────────────────────────────────────┘
                      ↓
┌──────────────────────────────────────────────┐
│              MODEL SERVING                   │
├──────────────────────────────────────────────┤
│ • Serving Platform: MLflow/Seldon/KServe     │
│ • Model Format: Serialized RandomForest (.pkl/.joblib) │
│ • API Framework: FastAPI/Flask with async support │
│ • Load Balancing: NGINX/HAProxy for high availability │
│ • Auto-scaling: Kubernetes HPA based on request volume │
└──────────────────────────────────────────────┘
                      ↓
┌──────────────────────────────────────────────┐
│          MONITORING & ALERTING               │
├──────────────────────────────────────────────┤
│ • Model Performance: Grafana dashboards with custom metrics│
│ • Data Drift Detection: Evidently AI/Great Expectations │
│ • System Health: Prometheus + AlertManager   │
│ • Logging: ELK Stack (Elasticsearch/Logstash/Kibana) │
│ • Alerting: PagerDuty/Slack integration      │
└──────────────────────────────────────────────┘
```

TECHNICAL SPECIFICATIONS

Infrastructure Requirements:
• Compute:
  - Training: 4 CPU cores, 16GB RAM (1-hour retraining)
  - Serving: 2 CPU cores, 4GB RAM (handles 1000 RPS)
  - Storage: 100GB SSD for data, models, and logs
• Network: 1Gbps bandwidth for real-time data ingestion
• Cloud: Multi-AZ deployment for 99.9% uptime SLA

Performance Characteristics:
• Prediction Latency: <50ms (p95), <20ms (p50)
• Throughput: 1000+ predictions/second per instance
• Memory Usage: 15MB model size + 500MB feature cache
• CPU Utilization: <30% under normal load
• Model Loading Time: <2 seconds (cold start)

API Specification:
```
POST /predict
Content-Type: application/json

Request Body:
{
  "timestamp": "2024-01-15T14:30:00Z",
  "features": {
    "current_trips": 15420,
    "hour": 14,
    "day_of_week": 1,
    "is_weekend": false
  }
}

Response:
{
  "prediction": 16250,
  "confidence_interval": [15800, 16700],
  "model_version": "v1.2.3",
  "prediction_id": "uuid-string",
  "timestamp": "2024-01-15T14:30:15Z"
}
```

Technical specifications prepared: September 12, 2025

RELIABILITY & SECURITY

High Availability Design:
• Multi-region deployment with active-passive failover
• Database replication with automated backups (RTO: 5min, RPO: 1min)
• Circuit breaker pattern for graceful degradation
• Blue-green deployment for zero-downtime updates
• Health checks with automatic instance replacement

Security Measures:
• API Authentication: JWT tokens with role-based access
• Data Encryption: TLS 1.3 in transit, AES-256 at rest
• Network Security: VPC with private subnets, security groups
• Audit Logging: All API calls logged with user attribution
• Compliance: SOC2 Type II, GDPR data protection standards

MONITORING & OBSERVABILITY

Key Performance Indicators:
• Business Metrics:
  - Prediction Accuracy: MAE < 500 trips (SLA)
  - API Availability: >99.9% uptime
  - Response Time: <100ms p95 latency
  - Data Freshness: <5 minute delay from source

• Technical Metrics:
  - Model Drift: Statistical tests on feature distributions
  - System Health: CPU, memory, disk, network utilization
  - Error Rates: 4xx/5xx HTTP responses <0.1%
  - Queue Depth: Message processing backlog <1000

Alert Configuration:
• Critical: Model accuracy drop >10% (immediate notification)
• Warning: API latency >200ms for >5 minutes
• Info: New model deployment completion
• Custom: Business-specific thresholds (peak hour accuracy)

CONTINUOUS IMPROVEMENT

Model Lifecycle Management:
• Automated Retraining: Weekly schedule with configurable triggers
• A/B Testing: Gradual rollout with statistical significance testing
• Model Versioning: Git-based version control with lineage tracking
• Performance Monitoring: Continuous validation against holdout set
• Rollback Strategy: Automatic fallback to previous version if degradation

Data Pipeline Optimization:
• Feature Engineering: Automated feature selection and engineering
• Data Quality: Anomaly detection and automated data cleaning
• Storage Optimization: Partitioning and compression strategies
• Caching Strategy: Multi-level caching for frequently accessed data

Operational Excellence:
• Runbook Documentation: Detailed troubleshooting guides
• Incident Response: Defined escalation procedures and contact lists
• Capacity Planning: Automated scaling based on demand forecasts
• Disaster Recovery: Cross-region backup and restoration procedures
• Training: Regular team training on system operations and updates

SCALABILITY PLANNING

Growth Projections:
• Current: 1M predictions/day
• 6 months: 5M predictions/day
• 1 year: 20M predictions/day
• 2 years: 100M predictions/day (multi-city expansion)

Scaling Strategy:
• Horizontal Scaling: Kubernetes auto-scaling with custom metrics
• Database Sharding: Time-based partitioning for historical data
• Caching: Multi-level cache architecture (L1: local, L2: Redis, L3: DB)
• CDN: Geographic distribution for global access
• Microservices: Decomposition for independent scaling of components

Technology Roadmap:
• Q1 2024: MLOps pipeline implementation
• Q2 2024: Real-time model updates and online learning
• Q3 2024: Multi-model ensemble deployment
• Q4 2024: Edge computing deployment for reduced latency
• 2025: Integration with IoT sensors and external data sources