Medical Data Analysis Pipeline - Documentation

Name: Alican
Surname: Sucu
Email: alicansucu@outlook.com
Project: Pusula Data Science Intern Case Study 2025

Table of Contents

Executive Summary

This document presents a comprehensive analysis of a physical medicine & rehabilitation dataset containing 2,235 patient records with 13 features. The primary objective was to conduct in-depth Exploratory Data Analysis (EDA) and prepare the data for potential predictive modeling with **TedaviSuresi** (Treatment Duration) as the target variable.

Key Achievements:

- ✅ Complete exploratory data analysis with statistical insights
- ✅ Comprehensive data preprocessing pipeline
- ✅ Feature engineering for complex categorical variables
- ✅ Data quality improvement and cleaning
- ✅ Model-ready dataset preparation

Dataset Overview

Dataset Characteristics

- **Total Records**: 2,235 observations
- **Features**: 13 columns
- **Domain**: Physical Medicine & Rehabilitation
- **Target Variable**: TedaviSuresi (Treatment Duration in Sessions)
- **Data Format**: Excel (.xlsx)

Column Specifications

Column	Type	Description	Missing Data Strategy
HastaNo	Identifier	Anonymized patient ID	Removed (not predictive)
Yas	Numerical	Patient age	Median imputation
Cinsiyet	Categorical	Gender	Label encoding
KanGrubu	Categorical	Blood type	Label encoding
Uyruk	Categorical	Nationality	Label encoding
KronikHastalik	Complex	Chronic conditions (comma-separated)	Binary feature extraction
Bolum	Complex	Department/Clinic (comma-separated)	Binary feature extraction
Alerji	Complex	Allergies (comma-separated)	Binary feature extraction
Tanilar	Text	Diagnoses	Removed (high correlation)
TedaviAdi	Text	Treatment name	Removed (high correlation)
TedaviSuresi	**Target**	**Treatment duration**	**Numeric conversion**
UygulamaYerleri	Text	Application sites	Removed (high correlation)
UygulamaSuresi	Numerical	Application duration	Numeric conversion

---

Exploratory Data Analysis Findings

1. Target Variable Analysis (TedaviSuresi)

Key Findings:

- **Data Type**: Initially stored as string, converted to numeric
- **Distribution**: Shows specific patterns in treatment session counts

- **Statistical Properties**:
  - Mean and median values indicate central tendency
  - Standard deviation reveals variability in treatment durations
  - Outliers detected using IQR method

Visualization Insights:

- Histogram reveals the frequency distribution of treatment sessions
- Box plot identifies outliers and quartile ranges
- Mean vs. median comparison indicates distribution skewness

2. Missing Data Analysis

Pattern Detection:

- Systematic analysis of missing data across all variables
- Missing data percentage calculated for each column
- Complete case analysis performed

Visualization Findings:

- **Heatmap**: Reveals missing data patterns and potential relationships
- **Bar Chart**: Shows missing data percentage by column
- **Impact Assessment**: Quantifies the effect of missing data on analysis

Key Statistics:

- Total missing values identified
- Complete rows percentage calculated
- Missing data distribution analyzed

3. Categorical Variables Analysis

Gender (Cinsiyet):

- Distribution analysis with count and percentage
- Class balance assessment
- Visualization: Pie chart for clear representation

Blood Type (KanGrubu):

- Distribution across different blood types
- Frequency analysis for each blood group
- Medical relevance assessment

Nationality (Uyruk):

- Patient demographic distribution
- Diversity analysis
- Potential impact on treatment patterns

4. Numerical Variables Analysis

Age (Yas):

- **Statistical Summary**: Mean, median, quartiles, min/max values
- **Distribution Analysis**: Histogram with mean and median lines
- **Outlier Detection**: IQR-based outlier identification
- **Age Groups**: Created categorical age brackets for analysis

Application Duration (UygulamaSuresi):

- **Correlation Analysis**: Relationship with treatment duration
- **Distribution Patterns**: Frequency analysis
- **Outlier Assessment**: Statistical outlier detection

5. Complex Categorical Analysis

Chronic Diseases (KronikHastalik):

- **Occupancy Rate**: Percentage of patients with chronic conditions
- **Top Conditions**: Most frequent chronic diseases identified
- **Feature Engineering**: Binary indicators for common conditions
- **Visualization**: Bar chart of top 10 most common conditions

Allergies (Alerji):

- **Allergy Patterns**: Distribution of different allergies
- **Frequency Analysis**: Most common allergic conditions
- **Medical Relevance**: Impact on treatment planning

Departments (Bolum):

- **Department Distribution**: Most active medical departments
- **Specialization Analysis**: Treatment patterns by department
- **Resource Allocation**: Insights for hospital management

6. Correlation Analysis

Numerical Correlations:

- Correlation matrix calculation for all numerical variables
- Heatmap visualization with color-coded correlation strengths
- Identification of strong positive/negative correlations
- Feature redundancy assessment

Data Preprocessing Steps

Step 1: Numeric Conversion

Purpose: Convert string-based numeric fields to proper numeric types

Process:

```python
# Extract numeric values using regex pattern (\d+)
df['TedaviSuresi_numeric'] = df['TedaviSuresi'].str.extract(r"(\d+)").astype(float)
df['UygulamaSuresi_numeric'] = df['UygulamaSuresi'].str.extract(r"(\d+)").astype(float)
```

Rationale: Original data stored as strings prevented numerical analysis

Step 2: Categorical Encoding

Purpose: Convert categorical variables to machine-readable format

Method: Label Encoding with missing value handling

```python
# Handle missing values with "Unknown" before encoding
le = LabelEncoder()
df[f"{col}_encoded"] = le.fit_transform(df[col].fillna("Unknown"))
```

Columns Processed:

- Cinsiyet → Cinsiyet_encoded
- KanGrubu → KanGrubu_encoded
- Uyruk → Uyruk_encoded

Step 3: Complex Categorical Processing

Purpose: Extract meaningful features from multi-value categorical fields

Methodology:

1. **Value Extraction**: Parse comma-separated values
2. **Frequency Analysis**: Count occurrences of each value
3. **Threshold Application**: Keep values appearing in ≥15% of records
4. **Binary Feature Creation**: Create has_[condition] indicators
5. **Count Features**: Create total_[category] count features

Example Implementation:

```python
# Create binary indicators for common conditions
for value in common_values:
    clean_name = value.replace(' ', '_')
    df[f'has_{clean_name}'] = df[col].fillna('').str.contains(value, case=False).astype(int)

# Create count features
df[f"total_{col}"] = df[col].fillna('').apply(
    lambda x: len([d.strip() for d in x.split(',') if d.strip()]) if x else 0
)
```

Step 4: Feature Engineering

Age Grouping:

```python
# Create age categories for analysis
df["Yas_Group"] = pd.cut(df["Yas"],
                        bins=[0, 18, 30, 50, 70, 100],
                        labels=[0, 1, 2, 3, 4],
                        include_lowest=True)
```

Age Categories:

- 0: 0-18 years (Pediatric)
- 1: 18-30 years (Young Adult)
- 2: 30-50 years (Middle-aged)
- 3: 50-70 years (Senior)
- 4: 70-100 years (Elderly)

Step 5: Missing Value Imputation

Strategy: Median imputation for numerical variables

```python
imputer = SimpleImputer(strategy="median")
df[numeric_cols] = imputer.fit_transform(df[numeric_cols])
```

Rationale:

- Median is robust to outliers
- Appropriate for medical data with potential extreme values
- Preserves distribution characteristics

Step 6: Feature Scaling

Method: StandardScaler (Z-score normalization)

```python
scaler = StandardScaler()
df[scale_cols] = scaler.fit_transform(df[scale_cols])
```

Columns Scaled:

- Yas (Age)
- UygulamaSuresi_numeric (Application Duration)

Purpose: Ensure features are on similar scales for machine learning

Step 7: Data Cleaning

Column Removal Rationale:

- **HastaNo**: Patient identifier, not predictive
- **Tanilar**: High correlation with target, prevents generalization
- **TedaviAdi**: Direct relationship to treatment duration
- **UygulamaYerleri**: Treatment-specific information
- **Original categorical columns**: Replaced by encoded versions

Duplicate Removal:

```python
# Remove duplicate rows
df.drop_duplicates(inplace=True)
```

Feature Engineering

Created Features Summary

Feature Type	Original Column	New Features	Purpose
Numeric	TedaviSuresi	TedaviSuresi_numeric	Target variable conversion
Numeric	UygulamaSuresi	UygulamaSuresi_numeric	Duration analysis
Categorical	Age	Yas_Group	Age-based segmentation
Binary	KronikHastalik	has_[condition]	Disease indicators
Count	KronikHastalik	total_KronikHastalik	Disease burden
Binary	Alerji	has_[allergy]	Allergy indicators
Count	Alerji	total_Alerji	Allergy count
Binary	Bolum	has_[department]	Department indicators
Count	Bolum	total_Bolum	Department complexity

Feature Selection Criteria

- **Frequency Threshold**: 15% minimum occurrence
- **Predictive Relevance**: Medical significance assessment
- **Data Quality**: Sufficient data availability
- **Independence**: Avoiding high correlation with target

Data Quality Assessment

Before Preprocessing

- **Shape**: Original dataset dimensions
- **Missing Data**: Identified missing value patterns
- **Duplicates**: Detected duplicate records
- **Data Types**: Mixed types requiring conversion
- **Inconsistencies**: String formats in numeric fields

After Preprocessing

- **Shape**: Final dataset dimensions
- **Completeness**: All missing values handled
- **Consistency**: Uniform data types
- **Feature Count**: Expanded feature set
- **Model Readiness**: Prepared for ML algorithms

Quality Metrics

- **Data Completeness**: 100% after imputation
- **Feature Consistency**: All numeric features properly scaled
- **Outlier Treatment**: Identified but preserved for medical relevance
- **Feature Validity**: All features have clear interpretation

---

Technical Implementation

Architecture Design

```
MedicalDataPipeline Class
├── __init__(): Initialize pipeline
├── load_data(): Excel file handling
├── basic_info(): Dataset overview
├── exploratory_data_analysis(): Complete EDA
├── preprocess_data(): Full preprocessing
├── data_quality_report(): Quality assessment
└── run_complete_pipeline(): End-to-end execution
```

Error Handling

- File format validation
- Missing column checks
- Data type verification
- Memory usage monitoring

Performance Considerations

- Efficient pandas operations
- Memory-optimized data structures
- Vectorized computations
- Minimal data copying

Key Insights and Recommendations

Medical Insights

1. **Treatment Patterns**: Specific patterns in treatment duration distribution
2. **Patient Demographics**: Age and gender distribution analysis
3. **Chronic Conditions**: Most common conditions affecting treatment
4. **Department Utilization**: Resource allocation insights

Data Quality Insights

1. **Missing Data**: Systematic patterns requiring attention
2. **Outliers**: Medical outliers vs. data quality issues
3. **Feature Relationships**: Strong correlations identified
4. **Data Consistency**: Improved through preprocessing

Modeling Recommendations

1. **Feature Selection**: Use engineered binary indicators
2. **Target Distribution**: Consider distribution characteristics
3. **Validation Strategy**: Account for medical domain knowledge
4. **Model Interpretation**: Ensure medical relevance

Future Enhancements

1. **Advanced Imputation**: KNN or iterative imputation
2. **Feature Selection**: Statistical significance testing
3. **Outlier Treatment**: Domain-specific outlier handling
4. **Model Integration**: Predictive model development

Methodology

EDA Approach

1. **Univariate Analysis**: Individual variable exploration
2. **Bivariate Analysis**: Relationship identification
3. **Multivariate Analysis**: Complex pattern detection
4. **Visual Analytics**: Comprehensive visualization strategy

Preprocessing Philosophy

1. **Data Integrity**: Preserve medical meaning
2. **Feature Engineering**: Create meaningful indicators
3. **Scalability**: Reusable pipeline design
4. **Documentation**: Clear methodology tracking

Validation Strategy

1. **Data Quality Checks**: Before/after comparison
2. **Feature Validation**: Medical relevance assessment
3. **Processing Verification**: Step-by-step validation
4. **Output Quality**: Model-readiness confirmation

Conclusion

This comprehensive analysis successfully transformed a raw medical dataset into a model-ready format suitable for predictive modeling. The pipeline addresses the key challenges of medical data including missing values, complex categorical variables, and data quality issues.

Achievements

- ✅ **Complete EDA**: Comprehensive statistical and visual analysis
- ✅ **Data Quality**: Improved consistency and completeness
- ✅ **Feature Engineering**: Created meaningful predictive features
- ✅ **Pipeline Design**: Modular, reusable architecture
- ✅ **Documentation**: Thorough methodology documentation

Dataset Transformation

- **Original**: 2,235 × 13 raw dataset
- **Processed**: Clean, feature-engineered dataset ready for ML
- **Features**: Expanded feature set with binary indicators
- **Quality**: 100% complete data with proper scaling

Business Impact

- **Treatment Insights**: Understanding of treatment patterns
- **Resource Planning**: Department utilization analysis
- **Quality Improvement**: Data-driven healthcare insights
- **Predictive Capability**: Foundation for treatment duration prediction

The pipeline successfully meets all requirements of the Pusula Data Science Intern Case Study 2025, providing a solid foundation for future predictive modeling work in the healthcare domain.

Document Version: 1.0
Last Updated: September 6, 2025
Project Status: Complete