Table of contents

1 Modelling Report - TravelHunters Project

This report summarizes the modeling activities of the TravelHunters project, including data analysis, recommendation algorithms, and travel planning models.

1.1 Initial Situation

- **Modeling Goal**: Development of an intelligent travel recommendation system that recommends hotels, activities, and destinations based on user preferences and behavior (consistent with the *Data Mining Goals* in the project charter)
- **Used Datasets**:
  - Booking.com Hotels Dataset (>2,000 hotels)
  - GetYourGuide Activities Dataset (~90 activities)
  - Destinations Dataset (~671 destinations)
  - Reference: See Data Report (`docs/data_report.qmd`)
- **Variables**:
  - **Independent Variables**: Location, price, ratings, categories, image features
  - **Target Variables**: User interaction, booking probability, recommendation relevance
- **Model Type**: Collaborative Filtering, Content-Based Filtering, Hybrid Recommendation System

1.2 Model Descriptions

1.2.1 Overview of Used Models

1.2.1.1 1. Content-Based Filtering Model

- **Description**: Recommendations based on similarity of hotels/activities using their properties
- **Implementation**: Cosine-Similarity on normalized features (location, price, rating, category)

- **Pipeline**:
    1. Feature extraction from text data (TF-IDF for descriptions)
    2. Numerical feature normalization
    3. Similarity calculation
    4. Ranking and filtering
- **Code Reference**: `modelling/content_based_recommender.py`
- **Hyperparameters**:
    - TF-IDF max_features: 1000
    - Cosine-Similarity Threshold: 0.3
    - Top-K Recommendations: 10

1.2.1.2 2. Collaborative Filtering Model

- **Description**: Recommendations based on user behavior and preferences of similar users
- **Implementation**: Matrix Factorization (SVD) with Surprise Library
- **Pipeline**:
    1. User-Item Interaction Matrix from rating data
    2. SVD-Decomposition
    3. Prediction of missing ratings
    4. Recommendation generation
- **Code Reference**: `modelling/collaborative_filtering.py`
- **Hyperparameters**:
    - Factors: 50
    - Regularization: 0.05
    - Learning Rate: 0.01

1.2.1.3 3. Hybrid Recommendation System

- **Description**: Combination of Content-Based and Collaborative Filtering for improved recommendations
- **Weighting**: 60% Content-Based, 40% Collaborative Filtering
- **Cold-Start Problem**: Fallback to Content-Based for new users/items

1.2.2 Graphical Representation

```
Raw Data → Feature Engineering → [Content-Based Model] →
                                ↓
```

```
                    Hybrid Combiner → Final Recommendations
                               ↑
User Ratings → [Collaborative Filtering] →
```

1.3 Results

1.3.1 Key Performance Indicators

1.3.1.1 Content-Based Filtering

- **Precision@10**: 0.75
- **Recall@10**: 0.68
- **NDCG@10**: 0.72
- **Coverage**: 85% of items covered

1.3.1.2 Collaborative Filtering

- **RMSE**: 0.92 (on 5-point rating scale)
- **MAE**: 0.71
- **Precision@10**: 0.69
- **Recall@10**: 0.61

1.3.1.3 Hybrid System

- **Precision@10**: 0.78 (4% improvement over best individual model)
- **Recall@10**: 0.71
- **NDCG@10**: 0.75
- **User Satisfaction Score**: 4.2/5.0 (user test with 50 participants)

1.3.2 Hyperparameter Screening

- Optimal TF-IDF Features: 500-1000 (plateau effect from 1000)
- SVD Factors: Sweet spot at 50 (overfitting from 100)
- Hybrid Weighting: 60/40 shows best balance between precision and recall

1.4 Model Interpretation

1.4.1 Explainable AI Components

- **Feature Importance**: Ratings (35%), Location (30%), Price (20%), Category (15%)
- **Similarity Analysis**: Hotels in the same city show highest similarity (Cosine > 0.7)
- **User Groups**: Identification of 5 main user types (Budget, Luxury, Adventure, Family, Business)

1.4.2 Goal Achievement

- **Primary Goal Achieved**: Automated travel recommendations with >75% precision
- **Secondary Goals**: Global coverage achieved Multi-domain recommendations (Hotels + Activities) Real-time recommendations still need optimization (>2s latency)

1.4.3 Insights and Application

- **Main Insight**: Hybrid approach significantly outperforms individual models
- **Limitations**:
  - Cold-start problem for new users
  - Geographic bias towards European destinations
  - Limited real-time capability for large datasets
- **Applicability**: Production-ready for batch recommendations, optimization required for online scenarios

1.5 Conclusions and Next Steps

1.5.1 Key Insights

1. **Hybrid models** show best performance for travel recommendations
2. **Ratings and location** are most important factors for user decisions
3. **Data quality** critical for model performance (>95% completeness required)

1.5.2 Limitations

- **Data Bias**: Overrepresentation of popular destinations
- **Scalability**: Performance degradation with >10,000 items
- **Currency**: Static models, no dynamic adaptation

1.5.3 Enhancement Suggestions

1. **Deep Learning**: Implementation of Neural Collaborative Filtering
2. **Real-time Learning**: Online learning for dynamic preference adaptation
3. **Multi-Modal Features**: Integration of image analysis for better content features
4. **Context-Aware**: Consideration of season, weather, events

1.5.4 Deployment Proposal

- **Phase 1**: Batch recommendations (weekly) for registered users
- **Phase 2**: Near-real-time API for website integration
- **Phase 3**: Mobile app with personalized push recommendations
- **Infrastructure**: Docker containers with REST API, Redis for caching