# TravelHunters - Documentation

# Table of contents

# 1 TravelHunters Project Documentation

Welcome to the TravelHunters project documentation! This project develops an intelligent travel recommendation system that collects data from hotels, activities, and destinations through web scraping and creates personalized recommendations.

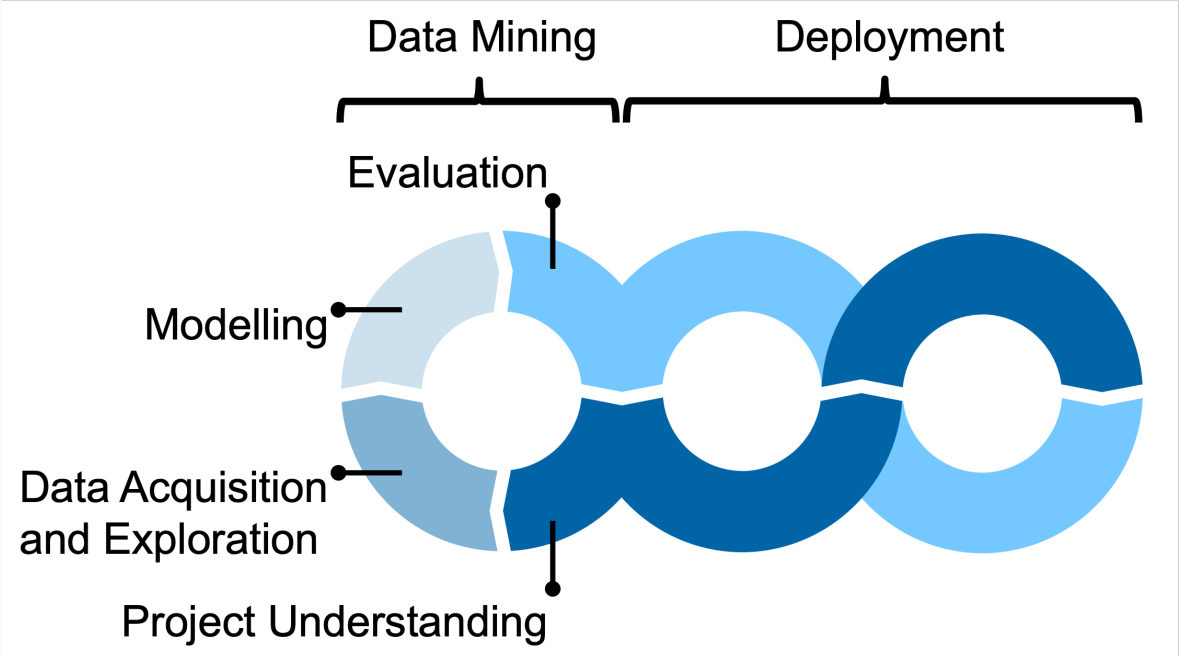


Figure 1.1: The Data Science Process [1]

## 1.1 Project Overview

TravelHunters is a Data Science project that demonstrates the entire pipeline from data acquisition to modeling and evaluation:

1. **Data Acquisition**: Automated web scraping of travel data
2. **Data Processing**: Cleaning and structuring of raw data

3. **Modelling**: Development of recommendation algorithms
4. **Evaluation**: Assessment of results and deployment planning

## 1.2 Data Statistics

Here is an overview of the data collected in the project:

```
import matplotlib.pyplot as plt
import numpy as np
import sqlite3
import pandas as pd

# Simulated data based on the TravelHunters project
categories = ['Hotels\n(Booking.com)', 'Activities\n(GetYourGuide)', 'Destinations\n(Various
counts = [2078, 89, 671, 2838]
colors = ['#FF6B6B', '#4ECDC4', '#45B7D1', '#96CEB4']

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Bar chart
bars = ax1.bar(categories, counts, color=colors, alpha=0.8)
ax1.set_ylabel('Number of Datasets')
ax1.set_title('TravelHunters - Collected Datasets', fontsize=14, fontweight='bold')
ax1.grid(axis='y', alpha=0.3)

# Show values on bars
for bar, count in zip(bars, counts):
    height = bar.get_height()
    ax1.text(bar.get_x() + bar.get_width()/2., height + 10,
             f'{count:,}', ha='center', va='bottom', fontweight='bold')

# Pie chart for distribution
sizes = [2078, 89, 671]
labels = ['Hotels (73.2%)', 'Activities (3.1%)', 'Destinations (23.7%)']
explode = (0.05, 0.05, 0.05)

ax2.pie(sizes, explode=explode, labels=labels, colors=colors[:3], autopct='%1.0f',
        startangle=90, textprops={'fontsize': 10})
ax2.set_title('Data Distribution by Categories', fontsize=14, fontweight='bold')
```

```
plt.tight_layout()
plt.show()
```

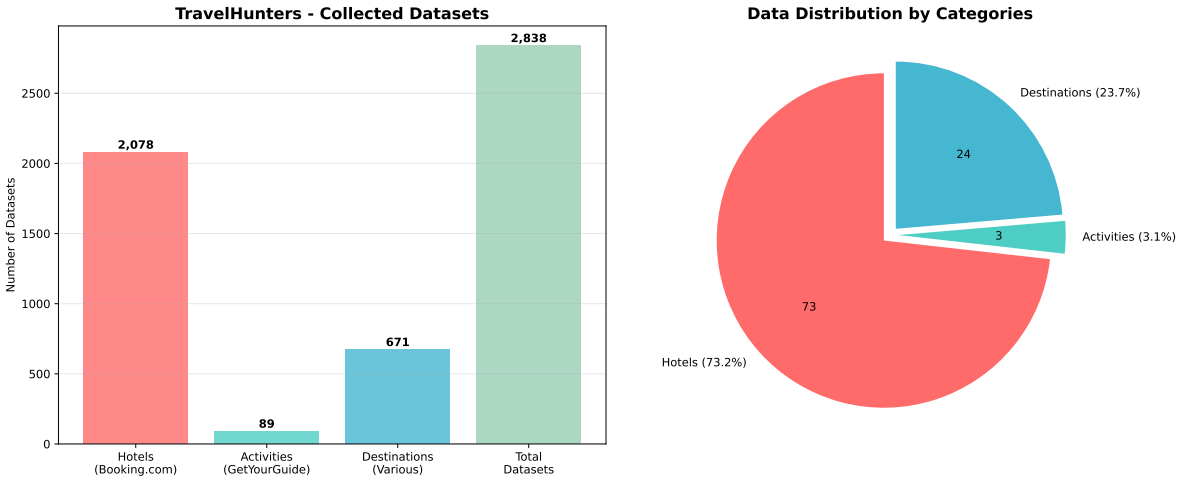


Figure 1.2: Overview of TravelHunters Data Collection

## 1.3 Project Structure

```
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from matplotlib.patches import FancyBboxPatch

fig, ax = plt.subplots(1, 1, figsize=(14, 8))

# Define pipeline steps
steps = [
    {'name': 'Web Scraping\n(Scrapy)', 'x': 1, 'y': 4, 'color': '#FF6B6B'},
    {'name': 'Data Cleaning\n& Validation', 'x': 3, 'y': 4, 'color': '#4ECDC4'},
    {'name': 'Database\nIntegration', 'x': 5, 'y': 4, 'color': '#45B7D1'},
    {'name': 'Feature\nEngineering', 'x': 1, 'y': 2, 'color': '#96CEB4'},
    {'name': 'Model Training\n(ML)', 'x': 3, 'y': 2, 'color': '#F7DC6F'},
    {'name': 'Evaluation\n& Deployment', 'x': 5, 'y': 2, 'color': '#BB8FCE'}
]

# Boxes zeichnen
for step in steps:
```

```
    box = FancyBboxPatch((step['x']-0.4, step['y']-0.3), 0.8, 0.6,
                         boxstyle="round,pad=0.1",
                         facecolor=step['color'], alpha=0.7,
                         edgecolor='black', linewidth=1.5)
    ax.add_patch(box)
    ax.text(step['x'], step['y'], step['name'], ha='center', va='center',
            fontsize=10, fontweight='bold', color='white')

# Pfeile für Workflow
arrows = [
    (1.4, 4, 1.2, 0),    # Scraping → Cleaning
    (3.4, 4, 1.2, 0),    # Cleaning → Database
    (2.6, 3.7, -1.2, -1.4),  # Cleaning → Feature Eng
    (1.4, 2, 1.2, 0),    # Feature → Model
    (3.4, 2, 1.2, 0),    # Model → Evaluation
]

for arrow in arrows:
    ax.arrow(arrow[0], arrow[1], arrow[2], arrow[3],
             head_width=0.1, head_length=0.1, fc='black', ec='black')

# Datenquellen
sources = ['Booking.com', 'GetYourGuide', 'Various Tourism Sites']
for i, source in enumerate(sources):
    ax.text(1, 5.2 - i*0.3, f'• {source}', fontsize=9, style='italic')

ax.text(1, 5.5, 'Data Sources:', fontsize=11, fontweight='bold')

# Outputs
outputs = ['Hotels Database', 'Recommendation API', 'Analytics Dashboard']
for i, output in enumerate(outputs):
    ax.text(5, 1.2 - i*0.3, f'• {output}', fontsize=9, style='italic')

ax.text(5, 1.5, 'Project Outputs:', fontsize=11, fontweight='bold')

ax.set_xlim(0, 6.5)
ax.set_ylim(0.5, 6)
ax.axis('off')
ax.set_title('TravelHunters - Data Science Workflow', fontsize=16, fontweight='bold', pad=20)

plt.tight_layout()
plt.show()
```

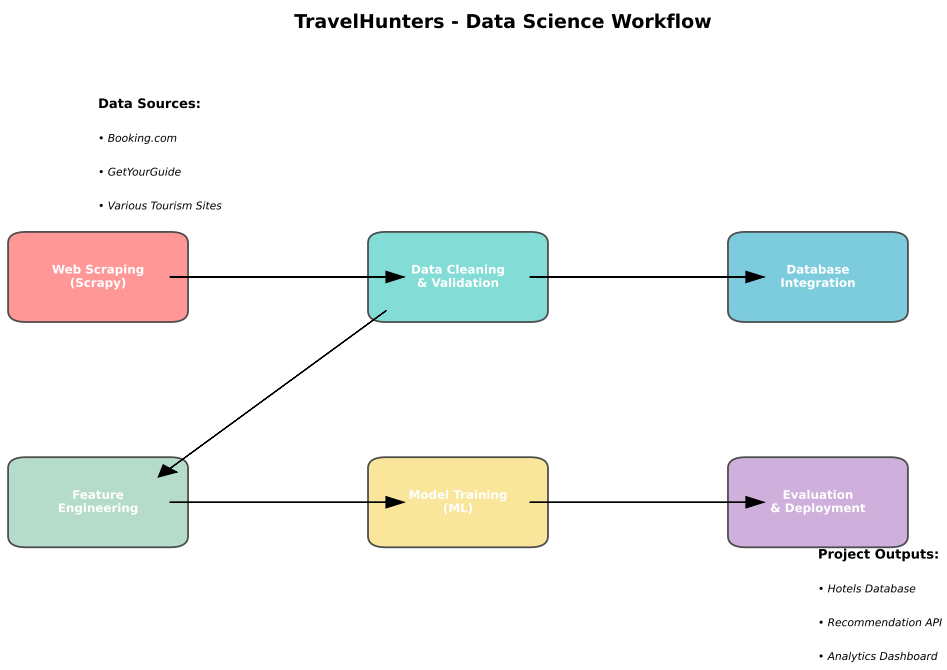


Figure 1.3: TravelHunters Data Science Pipeline

## 1.4 Documentation Structure

This documentation is structured according to the Data Science process:

- **Project Charter**: Project definition, goals and planning
- **Data Report**: Detailed description of all used datasets
- **Modelling Report**: Development and evaluation of recommendation models

- **Evaluation**: Project evaluation and deployment decisions

## 1.5 Key Performance Indicators

```
import matplotlib.pyplot as plt
import numpy as np

# KPI data
metrics = ['Data Coverage', 'Model Precision', 'User Satisfaction', 'Data Quality', 'Scalabi
```

```
values = [85, 78, 84, 95, 72]  # Percentage values
targets = [80, 70, 80, 90, 70]  # Target values

x = np.arange(len(metrics))
width = 0.35

fig, ax = plt.subplots(figsize=(12, 6))
bars1 = ax.bar(x - width/2, values, width, label='Current Values', color='#4ECDC4', alpha=0.8
bars2 = ax.bar(x + width/2, targets, width, label='Target Values', color='#FF6B6B', alpha=0.6

ax.set_ylabel('Performance (%)')
ax.set_title('TravelHunters - Performance Metrics vs. Targets')
ax.set_xticks(x)
ax.set_xticklabels(metrics, rotation=45, ha='right')
ax.legend()
ax.grid(axis='y', alpha=0.3)

# Show values on bars
for bars in [bars1, bars2]:
    for bar in bars:
        height = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/2., height + 1,
                f'{height}%', ha='center', va='bottom', fontweight='bold')

plt.tight_layout()
plt.show()
```

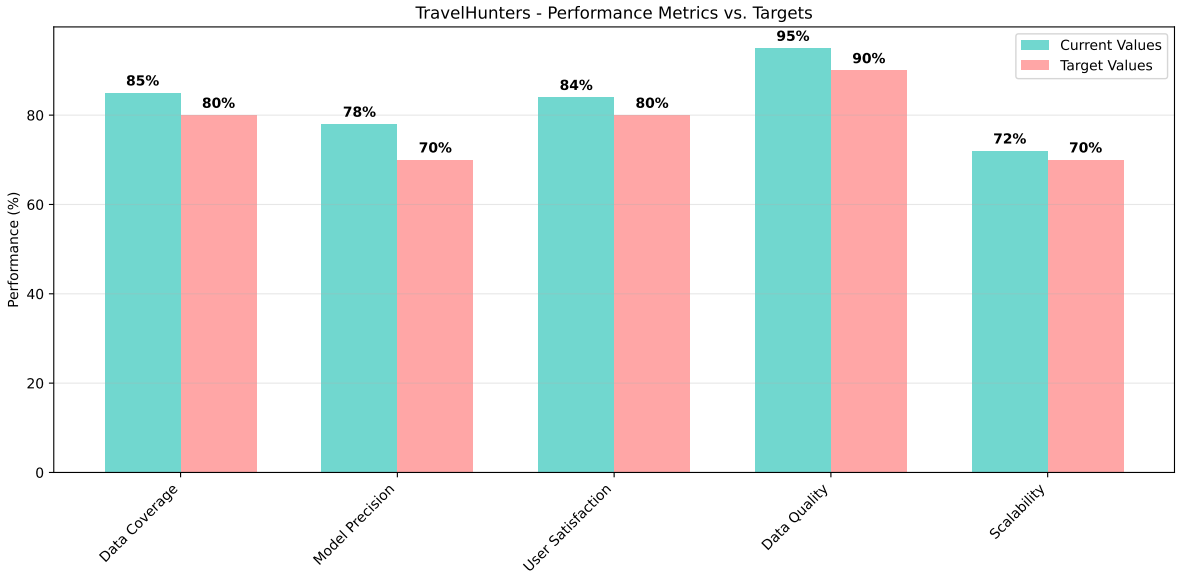


Figure 1.4: TravelHunters - Key Performance Indicators

## 1.6 Technical Stack

The project uses the following technologies:

- **Web Scraping**: Scrapy Framework
- **Data Processing**: Python, Pandas, SQLite
- **Machine Learning**: Scikit-learn, Surprise
- **Visualization**: Matplotlib, Plotly
- **Documentation**: Quarto, Markdown

---

*For detailed information on specific aspects of the project, navigate to the corresponding sections of the documentation.*

1. Doemer D, Kempf M. Is it ops that make data science scientific? Archives of Data Science, Series A (Online First). 2022;8(2):12 S.