Applied Data Science Semester Project 2025

Part A – Data Collection and Analysis

1. Approach Description

Initially, the Supreme Court website and the structure of its decisions were studied to determine the necessary specifications for the dataset.
- Developed a web crawler for data collection.
- Performed basic analysis and created visualizations for the results.
- Methodology based on course slides, with ChatGPT assisting in crawler implementation and regex creation.

2. Dataset Specifications

Based on the project requirements and the website structure, the following dataset specifications were defined.

Field	Type	Description
decision_number	String	Decision number
year	Int	Year of the decision (e.g., 2024)
department_type	String	Court department type
department_number	String	Department number
judges	List<String>	Judges involved in the decision
introduction_text	String	Introductory text of the decision
main_text	String	Main text of the decision
conclusion_text	String	Conclusion of the decision
penal_code	Set<String>	Articles of Penal Code (ΠΚ)
code_of_criminal_procedure	Set<String>	Articles of Code of Criminal Procedure (ΚΠΔ)
civil_code	Set<String>	Articles of Civil Code (ΑΚ)
code_of_civil_procedure	Set<String>	Articles of Code of Civil Procedure (Κπολδ)
decision_link	String	Link to the corresponding decision

3. Web Crawler

Python crawler developed (code in data_science_part1).
- Connected to the Supreme Court website using Selenium WebDriver.
- Discovered 2,475 article links and extracted information using BeautifulSoup and regex.

4. Results Analysis

- Successfully extracted decision number, year, department type, and department number.
- Text data and legal references were more challenging due to heterogeneous structures.
- Most articles follow a regex-detectable pattern; exceptions exist, causing some extraction errors.
- Legal references appear in multiple formats, complicating extraction using regular expressions.

Visualizations of the dataset are in data_science_part1.
Dataset link: decisions.csv

Part B – Legal Document Analysis

B1. Supervised Machine Learning for Document Classification

1. Setup

- Installed necessary libraries for preprocessing and model building.
- Implemented functions to load datasets for each label (volume, chapter, subject) and each Hugging Face set (train, validation, test).
- Functions were created for model construction and training.
- Multiple hyperparameter tests conducted; final models selected based on performance and computational efficiency.

2. Models and Hyperparameters

i) SVM

Label	Representation	C	Notes
volume	BoW	500	High performance
volume	TF-IDF	1000	High performance
chapter	BoW	1000	Slight decrease
chapter	TF-IDF	1000	Slight decrease
subject	BoW	1000	Lower performance due to few samples
subject	TF-IDF	1000	Lower performance due to few samples

ii) Logistic Regression

Label	C	Notes
volume	10	High performance
chapter	10	Performance decreases with more categories
subject	10	Performance lower due to few samples

iii) Random Forest

Label	n_estimators	max_depth	Notes
volume	100	20	Good performance
chapter	200	20	Decreased performance
subject	200	20	Decreased performance

Note: n > 500 caused MemoryError.
Observation: Many chapter and subject categories had very few samples, leading to zero precision, recall, and F1-score for these classes.

B2. Topic Analysis of Supreme Court Decisions

1. Data Preparation

- Data split into train (60%), validation (10%), test (20%).
- Exploratory analysis showed 315 unique categories; most frequent: 'Adequacy of reasoning'.
- Distribution is imbalanced, which may affect algorithms like K-Means.

2. K-Means Clustering

- Preprocessed using TF-IDF + SVD (500 features).
- K-Means applied with K = 2–20; evaluated using Macro/Micro Silhouette and NMI.

Text Type	Best K	Metric
Full Text	19	High NMI for case_category
Summary	20	Good NMI

Note: NMI preferred over Silhouette as it considers actual category labels.

3. LLM-based Title Extraction

- K-Means on summaries with K = 20 clusters.
- Selected three decisions per cluster (centroid-near or random) to create prompts for LLM.
- LLM (model: llama-4-maverick:free via OpenRouter) generated titles describing main legal issues.

Findings:
- Centroid-near selections produced clearer and more accurate titles.
- Random selections were often more general or less relevant.
- Conclusion: Centroid-based selection is more reliable for representing cluster topics.