

ELS - Checkpoint 1

João Félix - up202008867
João Malva - up202006605
Sofia Teixeira - up201806629

Implemented Features

- **Parsing Configuration:**
 - Uses YAML configuration files to define command sequences
- **Load**
 - Extracts data from YAML files
 - Assign a name to imported table
- **Rename:**
 - Renames columns based on a provided mapping
- **Save:**
 - Exports table data to a CSV file
 - Permits selection of columns

Semantic Model

- **Table Representation:**

- The Table class represents a table with columns and rows
- The table data can be manipulated through the commands

- **Commands:**

- Commands are actions that can be performed on tables
- Each command implements the Command interface and provides an execute() method
- At the moment three commands were implemented



Configuration File

```
 config.yaml ×  
1 - load:  
2     files:  
3         - "assignment1Files/decision_tree_1.yaml"  
4         - "assignment1Files/decision_tree_2.yaml"  
5     as: "some_table"  
6  
7 - rename:  
8     table: "some_table"  
9     columns:  
10    - classes_: "Classes"  
11    - splitter: "Splitter"  
12    - ccp_alpha: "CPP Alpha"  
13    - min_samples_split: "Min Samples Split"  
14  
15 - save:  
16     table: "some_table"  
17     columns: ["Criterion", "Splitter", "CPP Alpha", "Min Samples Split"]  
18     out: "resources/configurationFile.csv"
```

Parsing Process

- It uses a YAML parsing library (SnakeYAML) to parse the configuration file
- The Command interface defines a common method execute() that is added to a queue
- The commands are executed in the order they are defined in the configuration file

Design Decisions

- YAML
 - Readability
 - Flexibility
- Separation of Interfaces and Implementations
 - Maintainability
 - Uniform Execution
- Semantic Model
 - The semantic model, which includes the Table class and command classes, provides a clear representation of the data and operations that can be performed.

Demo





Questions?