

CSV開放資料處理

楊博旭 104590035

洪皓哲 103590044

Problem

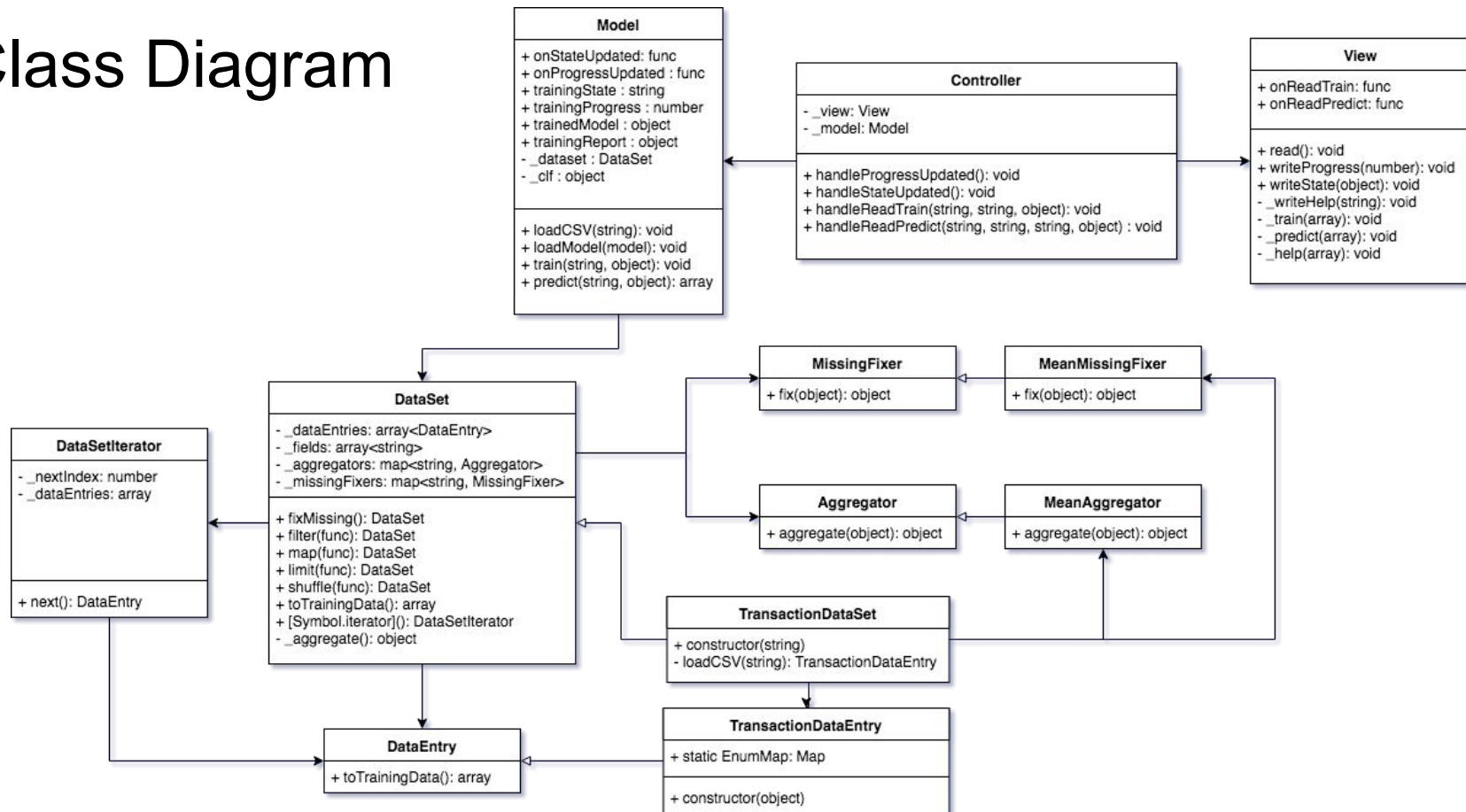
- Some of the fields is not machine learning friendly.
How to transform them?
- There are some missing parts in the data set.
How to fix them?

Design Problems

How to:

- separate the code of user interface and business logic?
- iterate through items in a custom data set class?
- chain multiple operations on a data set beautifully?

Class Diagram



Model View Controller

Why we use this pattern?

- Better code separation
- Simultaneous development

It solves our 1st design problem:

Separating code of user interface and business logic

Model View Controller

Consequence

- Need to write code more strictly
- Must concern more about role of the class
- Codes became much easier to manipulate

Model View Controller

Code Review

See:

- Model.js
- View.js
- Controller.js

Iterator

Why we use this pattern?

- Iterate through the items in our custom DataSet class
- And keep the array inside untouched

It solves our 2nd design problem:

Iterating through items in a custom data set class

Iterator

Consequence

- Iteration became more safer
(Other classes cannot modify the array object directly)

Iterator

Code Review

See:

- `DataSet.js`
- `DataSetIterator.js`

Strategy

Why we use this pattern?

- Reuse data handling algorithms on different data sets

Strategy

Consequence

We need to make strategies more:

- generic
- Simple

Strategy

Code Review

See:

- `aggregators/meanAggregator.js`
- `missingFixers/useMean.js`

Method Chaining

Why we use this pattern?

- Easier to apply multiple operations on same data set

Method Chaining

Consequence

- The method must return the object itself (not the result)
- Need to concern about side-effects

Method Chaining

Code Review

See:

- DataSet.js
 - filter(callback) : DataSet
 - map(callback) : DataSet
 - limit(number) : DataSet
 - shuffle() : DataSet
 - fixMissing() : DataSet

Problems Solved

Problems:

1. We can now transform fields that is not machine learning friendly. [use map()]
2. We can now fix missing parts in the data set. [use filter()]

Design Problems:

1. Separate the code of user interface and business logic - [MVC]
2. Iterating through items in a custom data set class - [Iterator]
3. Chaining multiple operations on a data set beautifully - [Method Chaining]