



# Allegato Tecnico

Gruppo TeamAFK - Progetto "Predire in Grafana"

gruppoafk15@gmail.com

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Approvatore</b>	Alessandro Canesso
<b>Redattori</b>	Victor Dutca
	Simone Meneghin
	Olivier Utshudi
	Davide Zilio
<b>Verificatori</b>	Simone Federico Bergamin
	Fouad Farid
	Simone Meneghin
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Cardin Riccardo
	TeamAFK

## Descrizione

*Allegato Tecnico* contenente le scelte architetturelle che il *TeamAFK* ha effettuato ai fini realizzativi del progetto *Predire in Grafana*. Comprende i design pattern utilizzati e i diagrammi di attività, sequenza, classi e package.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Riferimenti normativi	4
1.4.2	Riferimenti informativi	4
<b>2</b>	<b>Architettura del prodotto</b>	<b>5</b>
2.1	Descrizione generale	5
2.1.1	Diagrammi delle attività	6
2.2	Architettura Training Tool	11
2.2.1	Descrizione	11
2.2.2	Diagrammi dei package	12
2.2.3	Diagrammi delle classi	12
2.2.4	Diagrammi di sequenza	15
2.2.5	Design pattern notevoli utilizzati	15
2.3	Architettura Prediction Plug-in	15
2.3.1	Descrizione	15
2.3.2	Diagrammi dei package	15
2.3.3	Diagrammi delle classi	15
2.3.4	Diagrammi di sequenza	17
2.3.5	Design pattern notevoli utilizzati	17
<b>3</b>	<b>Requisiti soddisfatti</b>	<b>18</b>
3.1	Tabella del soddisfacimento dei requisiti	18
3.2	Grafici del soddisfacimento dei requisiti	21

## Elenco delle figure

2.1.1	Diagramma delle attività dello UC1 . . . . .	6
2.1.2	Diagramma delle attività dello UC2 . . . . .	7
2.1.3	Diagramma delle attività dello UC3 . . . . .	8
2.1.4	Diagramma delle attività dello UC4 . . . . .	9
2.1.5	Diagramma delle attività dello UC5 . . . . .	10
2.1.6	Diagramma delle attività dello UC6 . . . . .	11
2.2.1	Diagramma dei package del Training Tool . . . . .	12
2.2.2	Diagramma delle classi del Model del Training Tool . . . . .	12
2.2.3	Diagramma delle classi della View del Training Tool . . . . .	13
2.2.4	Diagramma delle classi del ViewModel del Training Tool . . . . .	14
2.2.5	Diagramma di sequenza del TrainSVM . . . . .	15
2.3.1	Diagramma dei package del Prediction Plug-in . . . . .	15
2.3.2	Diagramma delle classi del Model del Prediction Plug-in . . . . .	16
2.3.3	Diagramma delle classi della View del Prediction Plug-in . . . . .	16
2.3.4	Diagramma delle classi del Controller del Prediction Plug-in . . . . .	17

## Elenco delle tabelle

<a href="#">3.1.1 Tabella del soddisfacimento dei requisiti . . . . .</a>	18
---	----

# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo del documento è una descrizione esaustiva delle capacità del software *predire in Grafana* sviluppato dal team AFK. Il documento si concluderà con un resoconto su quanto sia stato soddisfatto dei vari requisiti.

## 1.2 Scopo del prodotto

Predire in Grafana<sub>G</sub> soddisfa le necessità di monitorare costantemente applicazioni e informazioni contenute in esse. Con questo scopo il team AFK si propone per la realizzazione per l'azienda Zucchetti S.p.A. di un tool<sub>G</sub> di addestramento e di un plug-in<sub>G</sub> di monitoraggio per Grafana che utilizzi algoritmi di SVM<sub>G</sub> e Regressione Lineare<sub>G</sub> sul dati in ingresso.

## 1.3 Glossario

Per evitare ambiguità nei documenti formali, viene fornito il documento *Glossario*, contenente tutti i termini considerati di difficile comprensione. Perciò nella documentazione fornita ogni vocabolo contenuto nel Glossario è contrassegnato dalla lettera G a pedice.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- **Capitolato Appalto C4** : <https://www.math.unipd.it/tullio/IS-1/2019/Progetto/C4.pdf>
- Norme di progetto v3.0.0
- Aggiungere verbali

### 1.4.2 Riferimenti informativi

Analisi di requisiti v3.0.0

## 2 Architettura del prodotto

### 2.1 Descrizione generale

Il progetto *Predire in Grafana* prevede la realizzazione di due moduli: un plug-in per la piattaforma Grafana e un tool esterno di supporto, rispettivamente chiamati **Prediction Plug-in** e **Training Tool**.

Il Training Tool si occupa di addestrare un algoritmo di *SVM* o *Regressione Lineare* utilizzando un dataset inserito dall'utente, per poi generare un file json contenente le informazioni necessarie per poter effettuare un calcolo di predizione. Questo modulo è stato sviluppato seguendo il pattern *Model-View-ViewModel (MVVM)*.

Il Prediction Plug-in invece si occuperà di ricevere in input il json e una volta collegati i predittori contenuti nel file ad un flusso dati, permetterà di iniziare ad effettuare i calcoli di previsione. Questo modulo è stato sviluppato seguendo il pattern *Model-View-Controller (MVC)*.

Le motivazioni principali che hanno portato alla scelta del design pattern MVVM per il Training Tool sono:

- per la realizzazione del componente è stato utilizzato *React* e abbiamo ritenuto che questo pattern si accoppiasse bene con la struttura di *React*;

Le motivazioni principali che hanno portato alla scelta del design pattern MVC per il Prediction Plug-in sono:

- abbiamo ritenuto che questo pattern si accoppiasse meglio con la struttura dei plug-in di Grafana;

Inoltre entrambi i pattern permettono:

- di disaccoppiare la parte di *presentation logic* da quella di *business logic*;
- il riutilizzo di alcune componenti in altri contesti.

### 2.1.1 Diagrammi delle attività

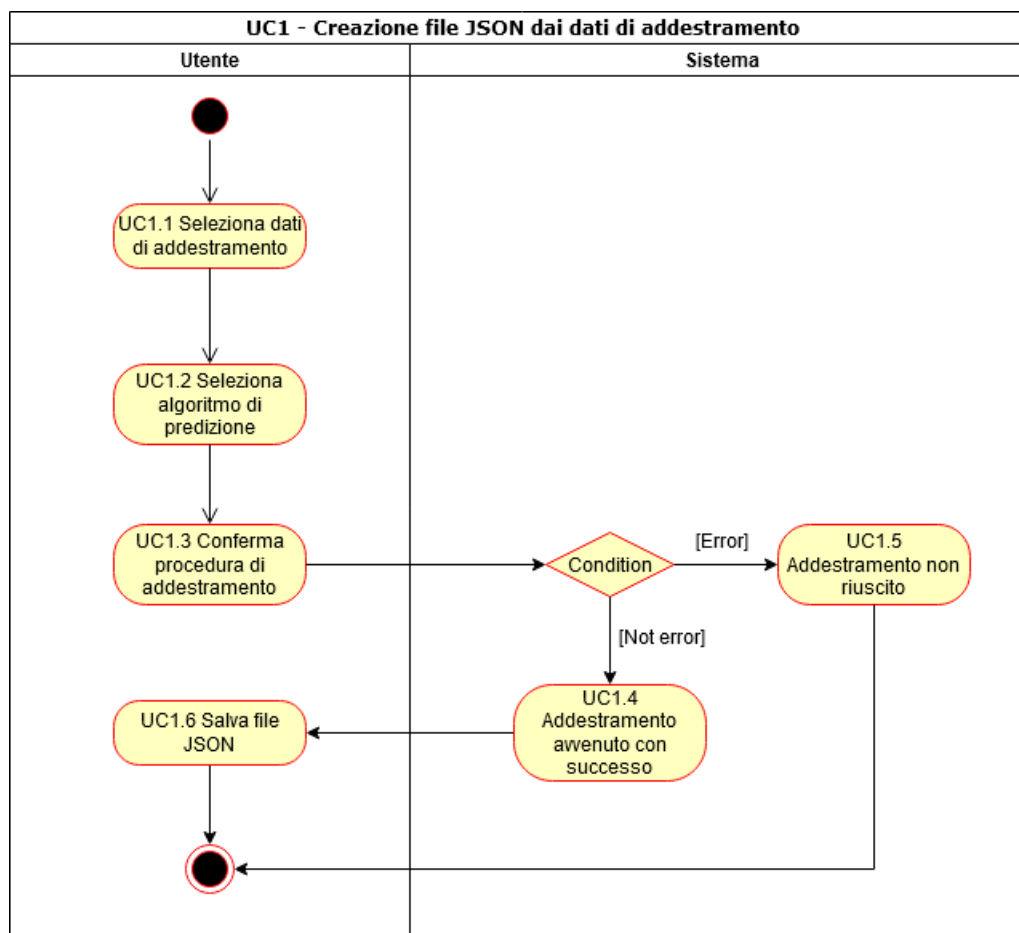


Figura 2.1.1: Diagramma delle attività dello UC1

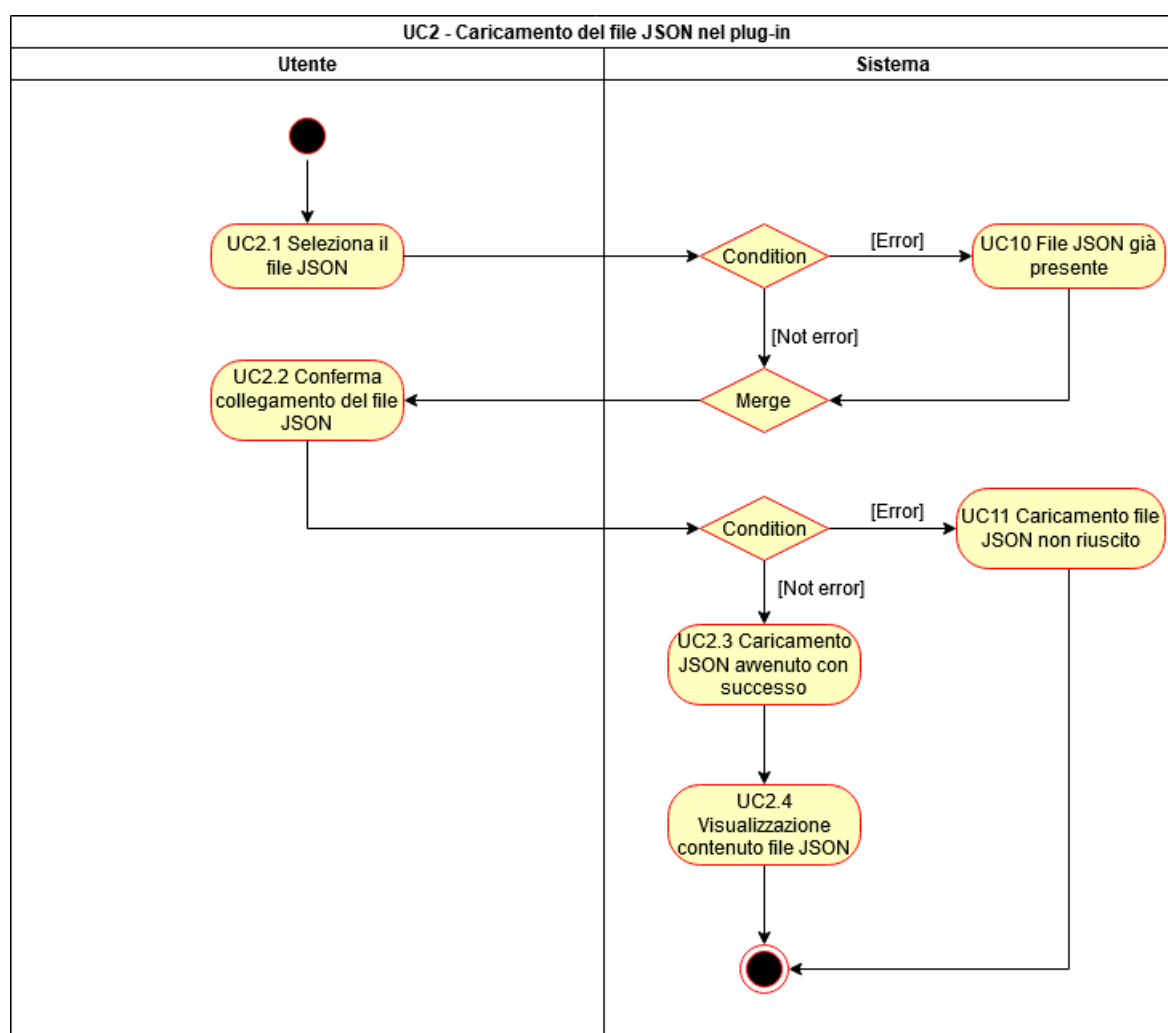


Figura 2.1.2: Diagramma delle attività dello UC2



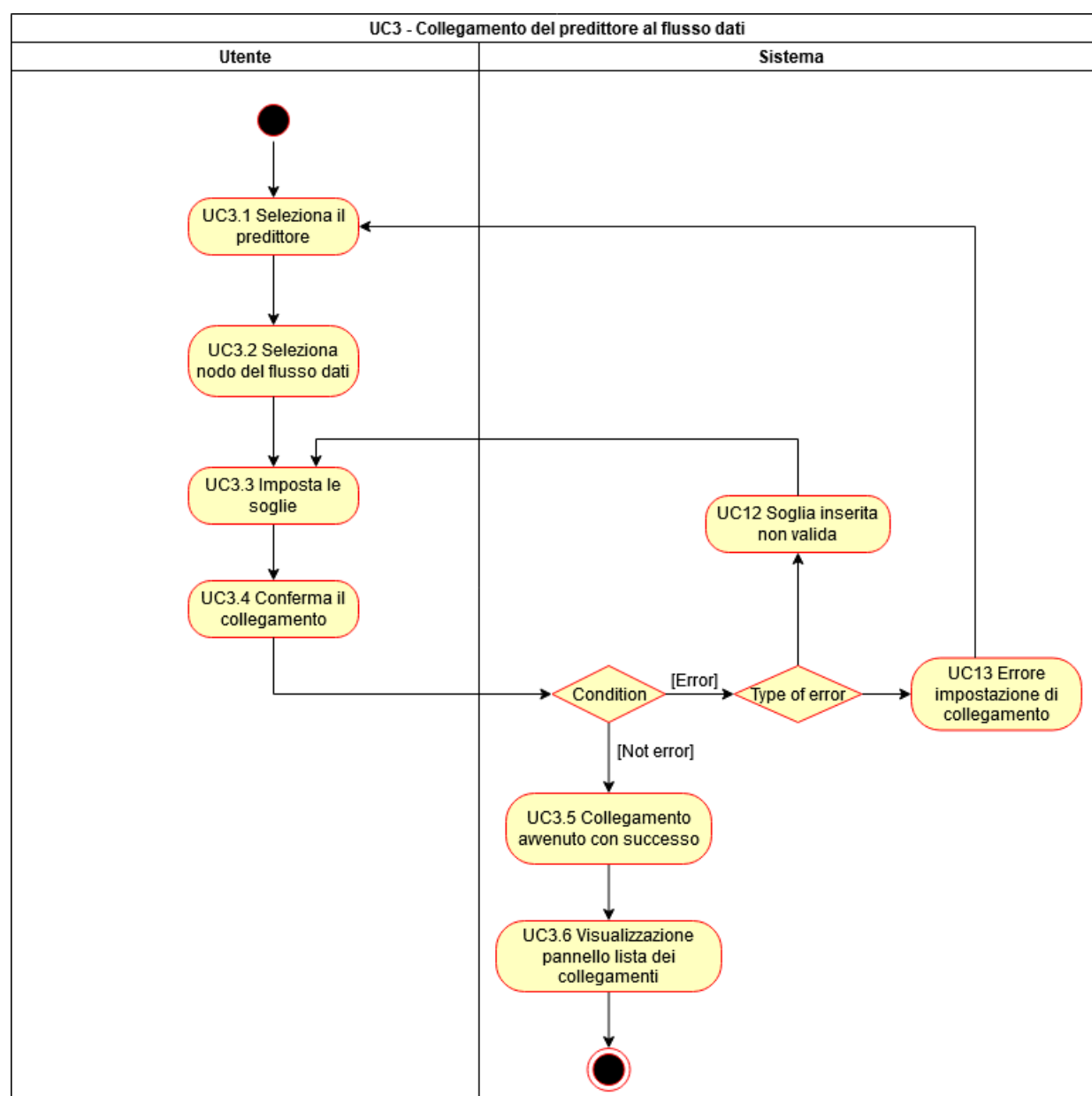


Figura 2.1.3: Diagramma delle attività dello UC3

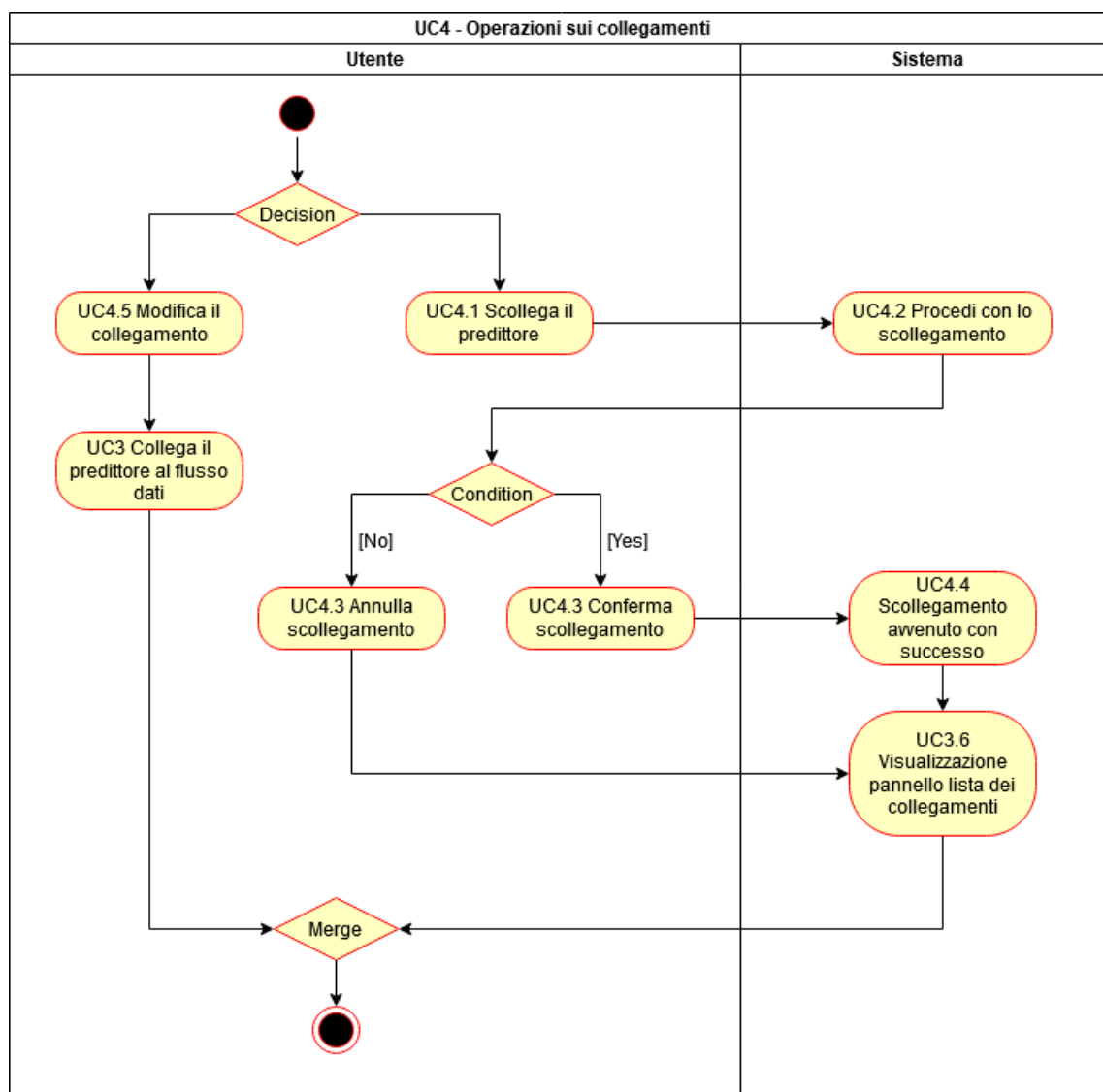


Figura 2.1.4: Diagramma delle attività dello UC4

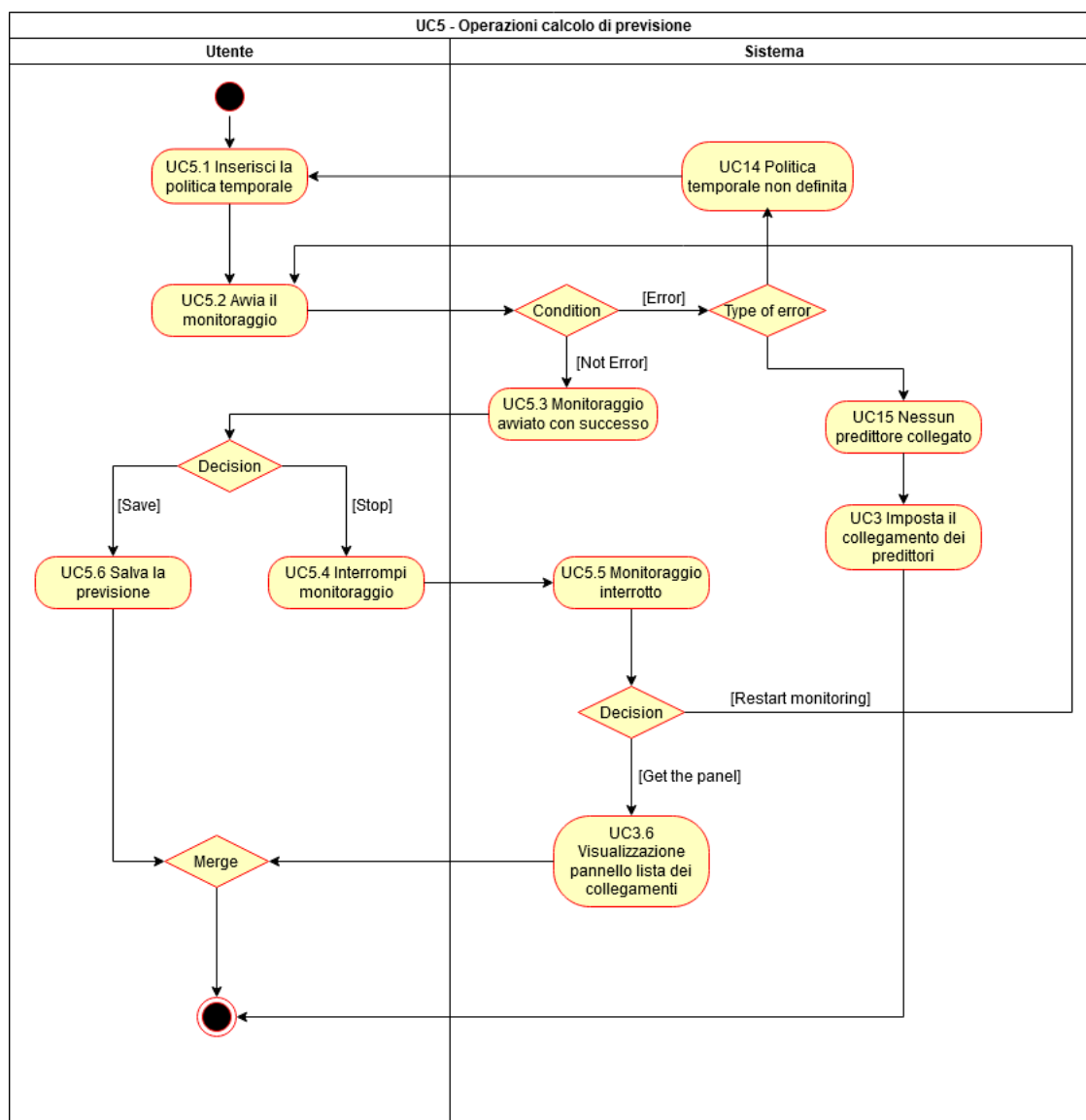


Figura 2.1.5: Diagramma delle attività dello UC5

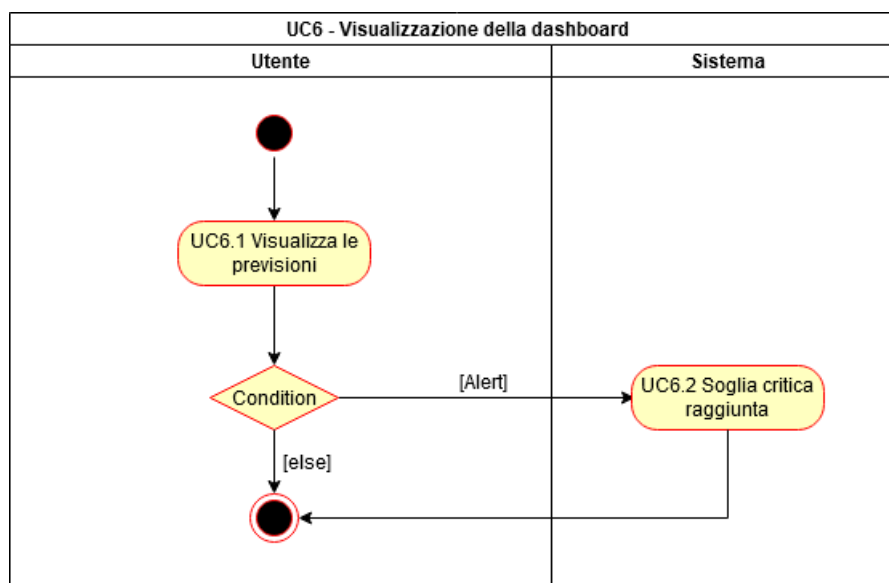


Figura 2.1.6: Diagramma delle attività dello UC6

## 2.2 Architettura Training Tool

### 2.2.1 Descrizione

Come specificato in precedenza, in questo modulo è stato utilizzato il pattern architetturale MVVM. Il passaggio di dati da Model alle varie View avviene tramite la modifica di una variabile props iniettata dal Controller (ModelView). Il Controller passa tramite queste props i giusti metodi da chiamare quando l'utente interagisce con la vista. Questo utilizzo delle props favorisce la separazione tra la presentation logic e la business logic. Per l'aggiornamento dei dati riferiti al Model e la notifica del cambiamento di questi dati al Controller abbiamo utilizzato la libreria MobX. Essa ci ha permesso di implementare il meccanismo degli Observer non nativamente supportato in React.

Per quanto riguarda la scelta effettuata dall'utente, in merito all'algoritmo da utilizzare, verrà:

- istanziata la Concrete Strategy corretta tramite il giusto indice di un array che ne contiene tutte le possibili scelte;
- visualizzata la specifica vista relativa all'algoritmo selezionato, sempre tramite un array contenente le possibili scelte.

## 2.2.2 Diagrammi dei package

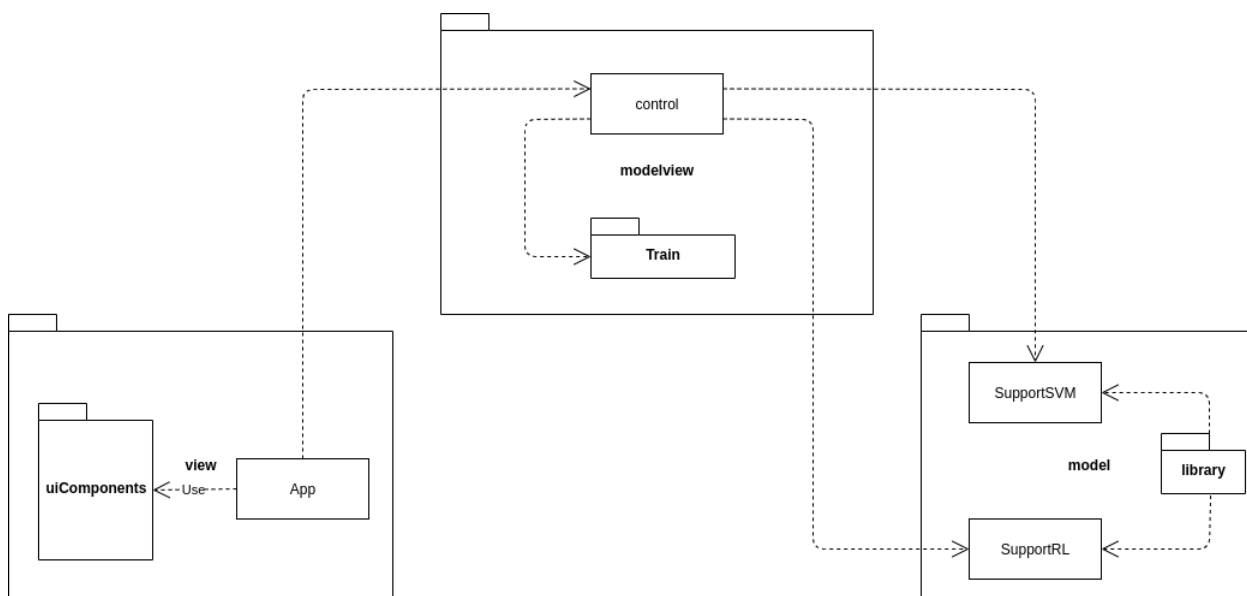


Figura 2.2.1: Diagramma dei package del Training Tool

## 2.2.3 Diagrammi delle classi

### Model

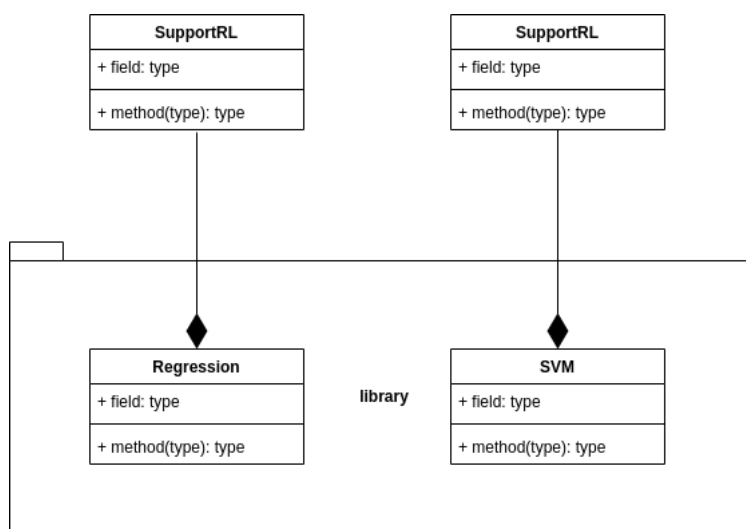


Figura 2.2.2: Diagramma delle classi del Model del Training Tool

### View

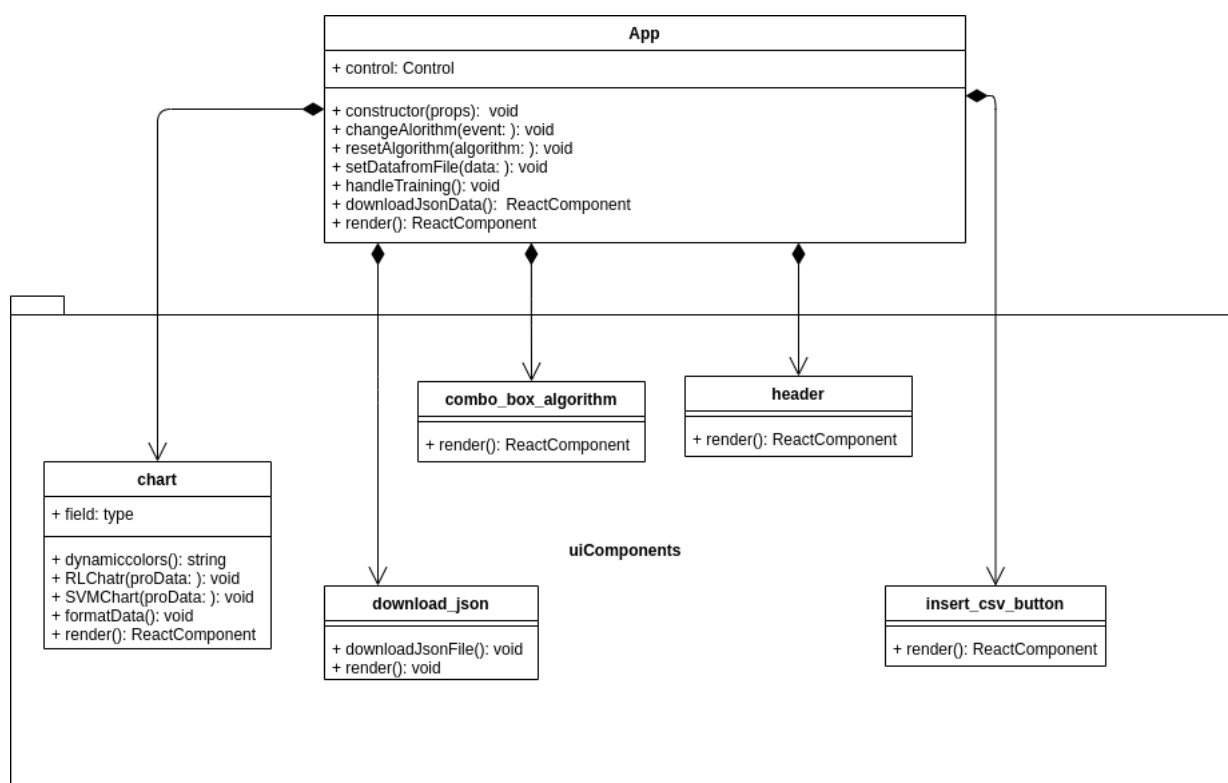


Figura 2.2.3: Diagramma delle classi della View del Training Tool

## ViewModel

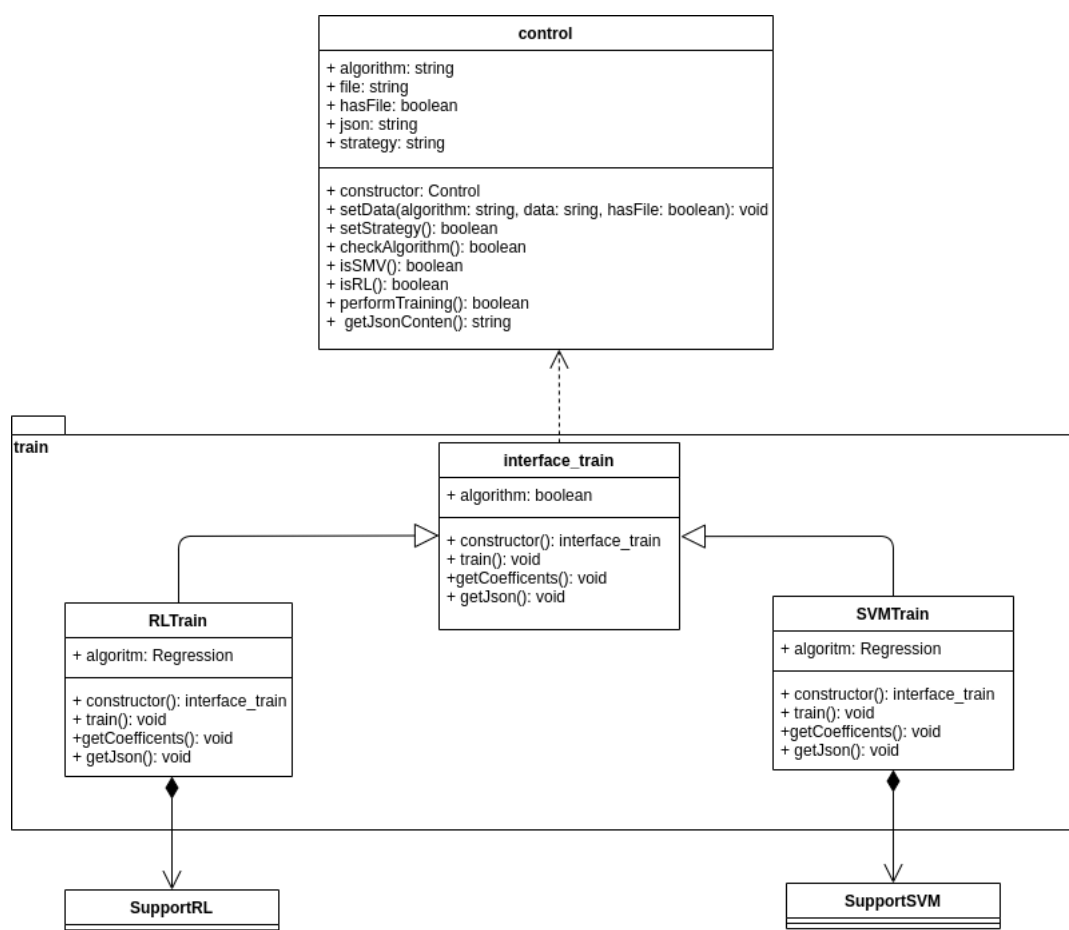


Figura 2.2.4: Diagramma delle classi del ViewModel del Training Tool

## 2.2.4 Diagrammi di sequenza

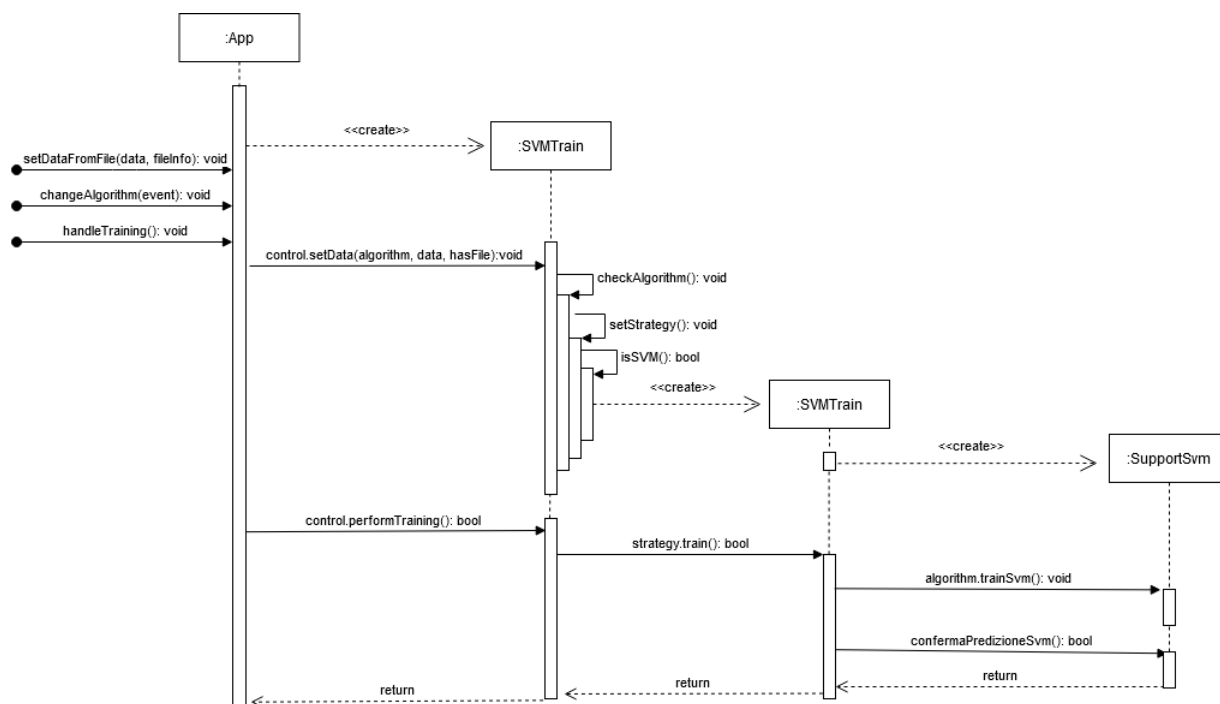


Figura 2.2.5: Diagramma di sequenza del TrainSVM

## 2.2.5 Design pattern notevoli utilizzati

## 2.3 Architettura Prediction Plug-in

### 2.3.1 Descrizione

### 2.3.2 Diagrammi dei package

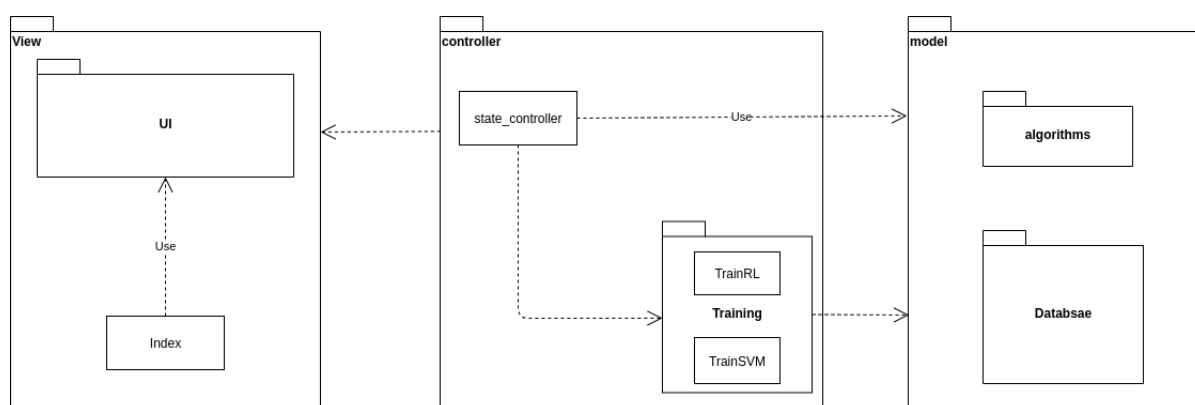


Figura 2.3.1: Diagramma dei package del Prediction Plug-in

### 2.3.3 Diagrammi delle classi

#### Model



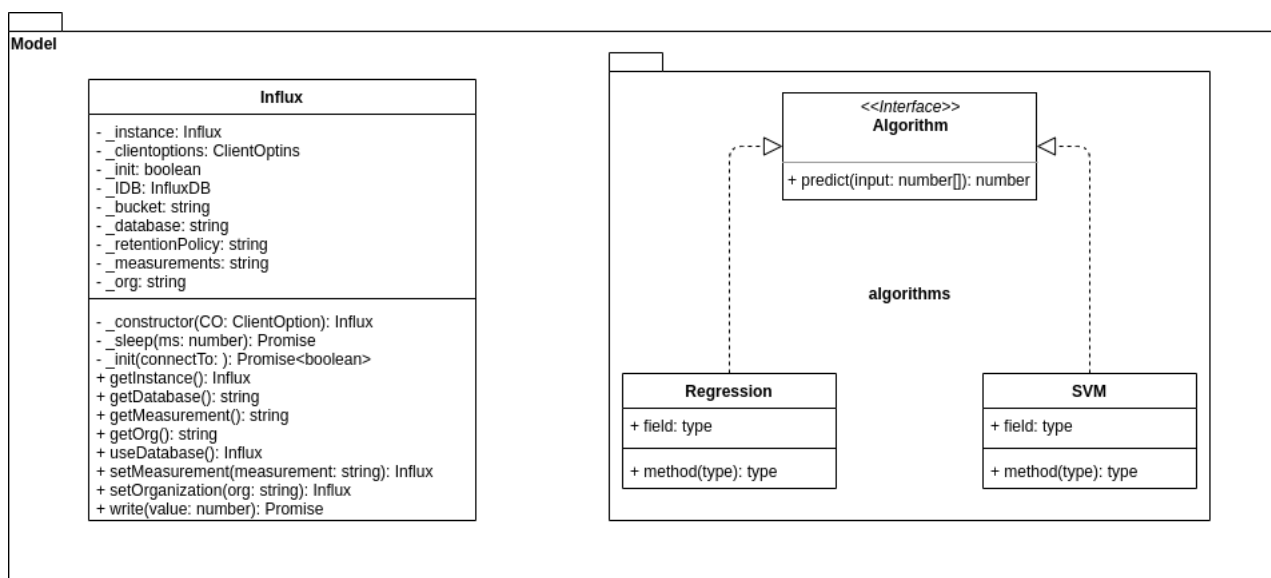


Figura 2.3.2: Diagramma delle classi del Model del Prediction Plug-in

## View

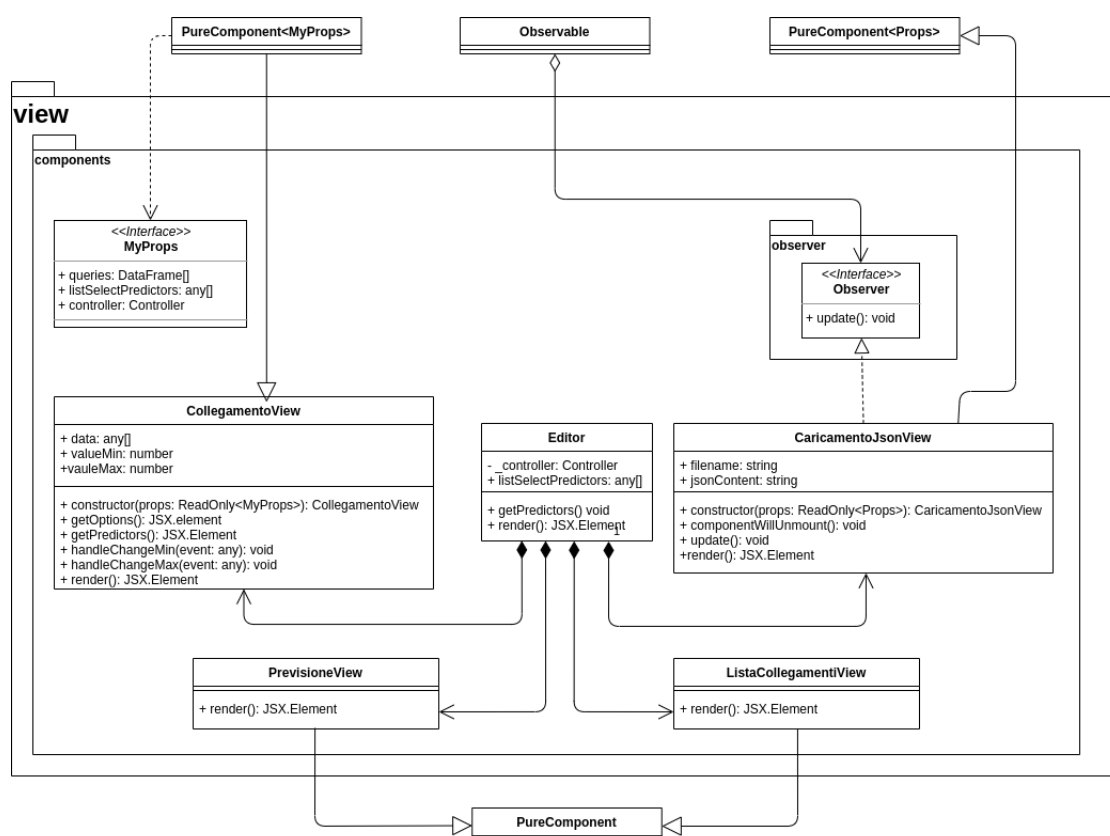


Figura 2.3.3: Diagramma delle classi della View del Prediction Plug-in

## Controller

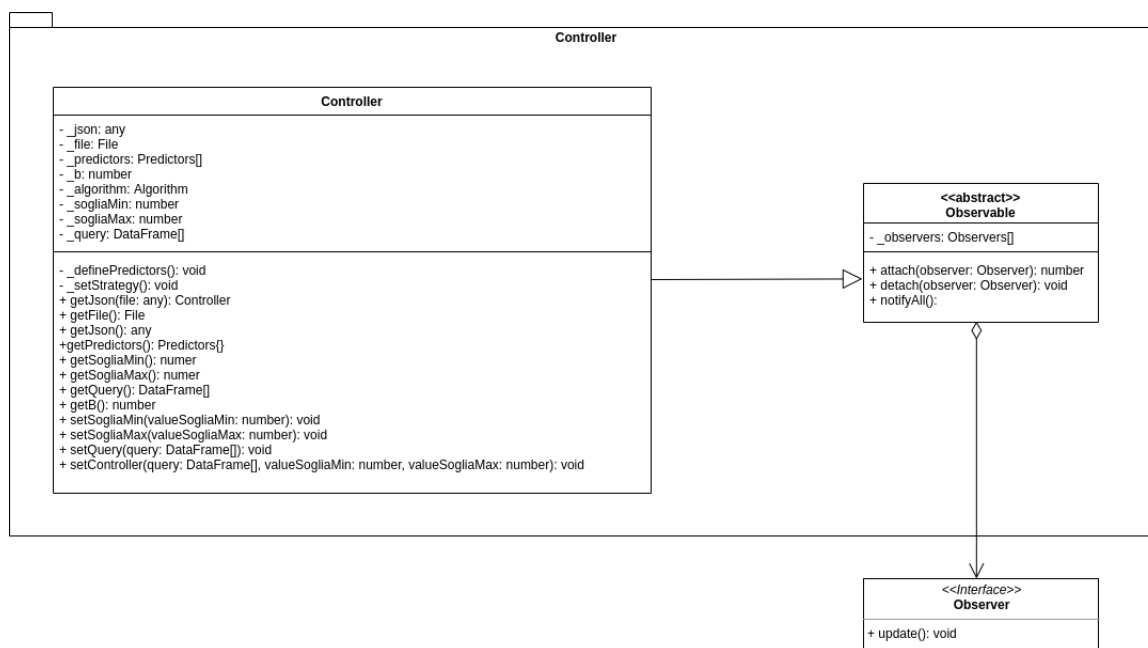


Figura 2.3.4: Diagramma delle classi del Controller del Prediction Plug-in

### 2.3.4 Diagrammi di sequenza

### 2.3.5 Design pattern notevoli utilizzati

## 3 Requisiti soddisfatti

### 3.1 Tabella del soddisfacimento dei requisiti

Tabella 3.1.1: Tabella del soddisfacimento dei requisiti

Codice	Esito
Re1F1	Soddisfatto
Re1F1.1	Soddisfatto
Re1F1.2	Soddisfatto
Re1F1.3	Soddisfatto
Re1F1.4	Soddisfatto
Re1F2	Soddisfatto
Re1F2.1	Soddisfatto
Re1F2.2	Soddisfatto
Re1F3	Soddisfatto
Re1F3.1	Soddisfatto
Re1F3.2	Soddisfatto
Re1F3.3	Soddisfatto
Re1F3.4	Soddisfatto
Re1F4	Soddisfatto
Re1F5	Soddisfatto
Re1F6	Soddisfatto
Re1F6.1	Soddisfatto
Re1F6.2	Soddisfatto
Re1F6.3	Soddisfatto
Re1F7	Soddisfatto
Re1F8	Soddisfatto
Re1F9	Soddisfatto
Re1F10	Soddisfatto
Re1F11	Soddisfatto
Re1F12	Soddisfatto

Tabella 3.1.1: (continua)

Codice	Esito
Re1F13	Soddisfatto
Re1F14	Soddisfatto
Re1F15	Soddisfatto
Re1F16	Soddisfatto
Re1F17	Soddisfatto
Re1F18	Soddisfatto
Re1F19	Soddisfatto
Re1F20	Soddisfatto
Re1F21	Soddisfatto
Re1F22	Soddisfatto
Re1F23	Soddisfatto
Re3F24	Soddisfatto
Re1Q1	Soddisfatto
Re1Q2	Soddisfatto
Re1Q2.1	Soddisfatto
Re2Q2.2	Soddisfatto
Re1Q3	Soddisfatto
Re1Q4	Soddisfatto
Re2Q5	Soddisfatto
Re2Q6	Soddisfatto
Re2Q7	Soddisfatto
Re1V1	Soddisfatto
Re1V1.1	Soddisfatto
Re1V1.2	Soddisfatto
Re1V1.3	Soddisfatto
Re1V1.4	Soddisfatto
Re1V2	Soddisfatto
Re1V3	Soddisfatto

Tabella 3.1.1: (continua)

Codice	Esito
Re1V4	Soddisfatto
Re1V5	Soddisfatto

## 3.2 Grafici del soddisfacimento dei requisiti