



Allegato Tecnico

Gruppo TeamAFK - Progetto "Predire in Grafana"

gruppoafk15@gmail.com

Informazioni sul documento

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

Descrizione

Allegato Tecnico contenente le scelte architettureali 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

1	Introduzione	4
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
2	Architettura del prodotto	5
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
3	Requisiti soddisfatti	18
3.1	Tabella del soddisfacimento dei requisiti	18
3.2	Grafici del soddisfacimento dei requisiti	18

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

3.1.1 Tabella del soddisfacimento dei requisiti	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_G 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_G di addestramento e di un plug-in_G di monitoraggio per Grafana che utilizzi algoritmi di SVM_G e Regressione Lineare_G 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

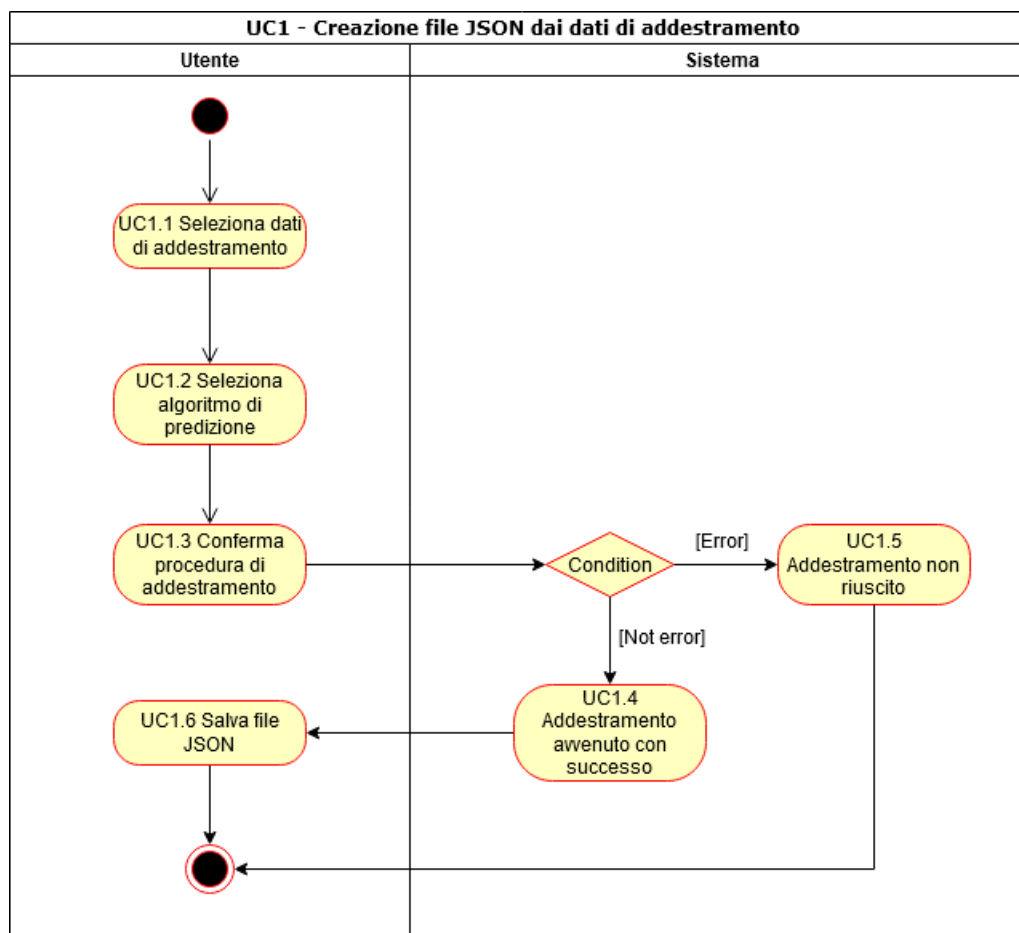


Figura 2.1.1: Diagramma delle attività dello UC1

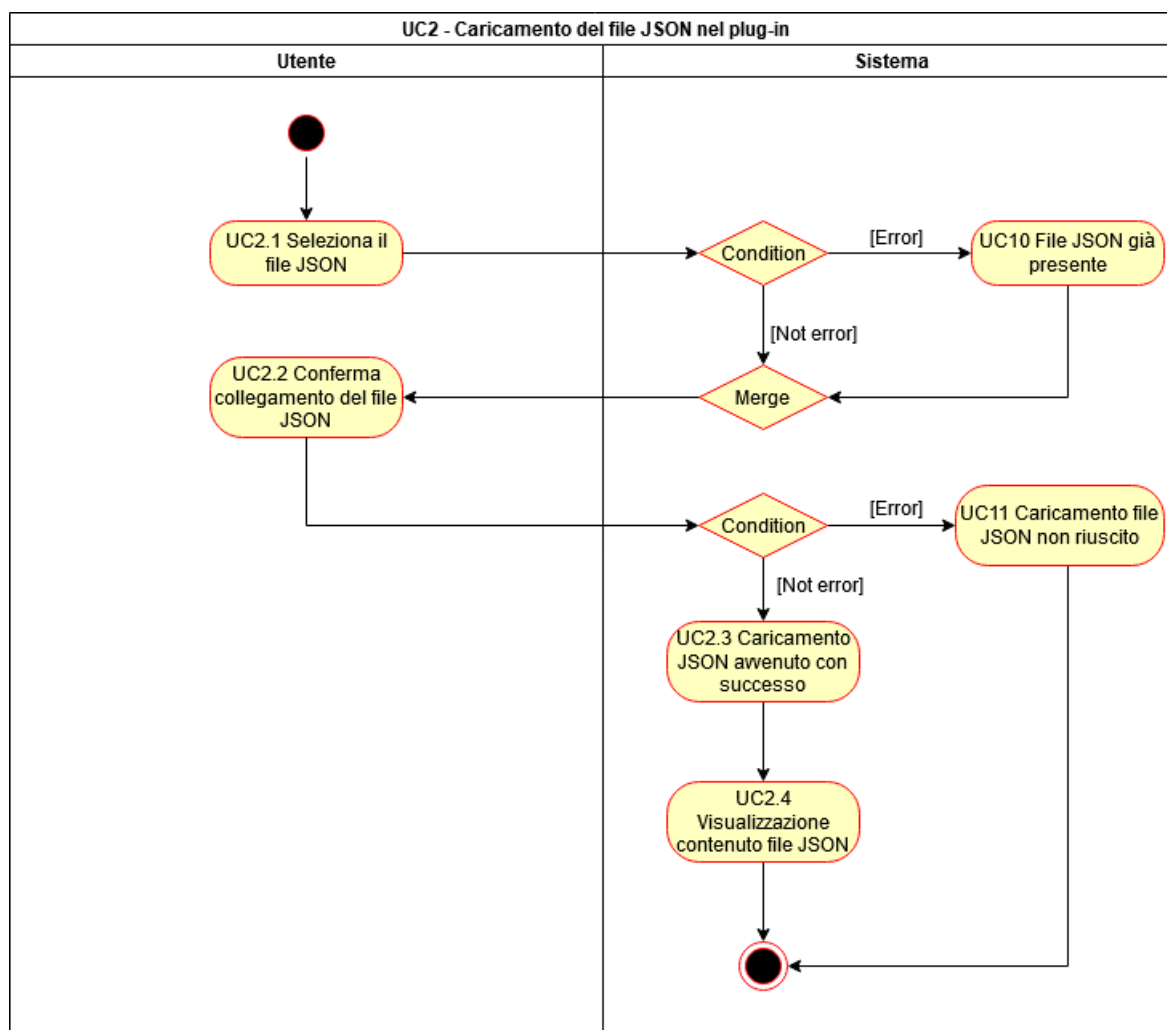


Figura 2.1.2: Diagramma delle attività dello UC2

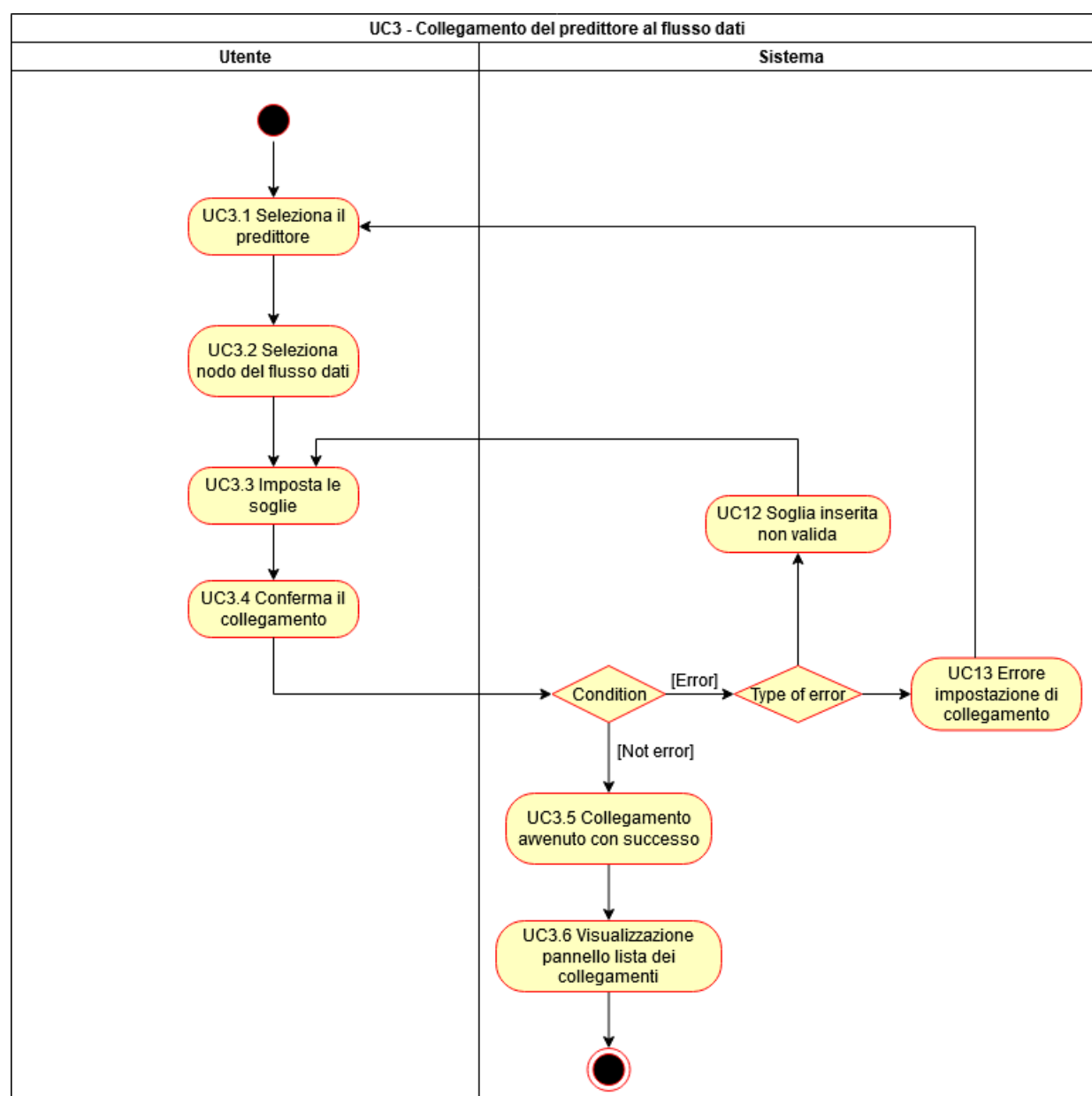


Figura 2.1.3: Diagramma delle attività dello UC3

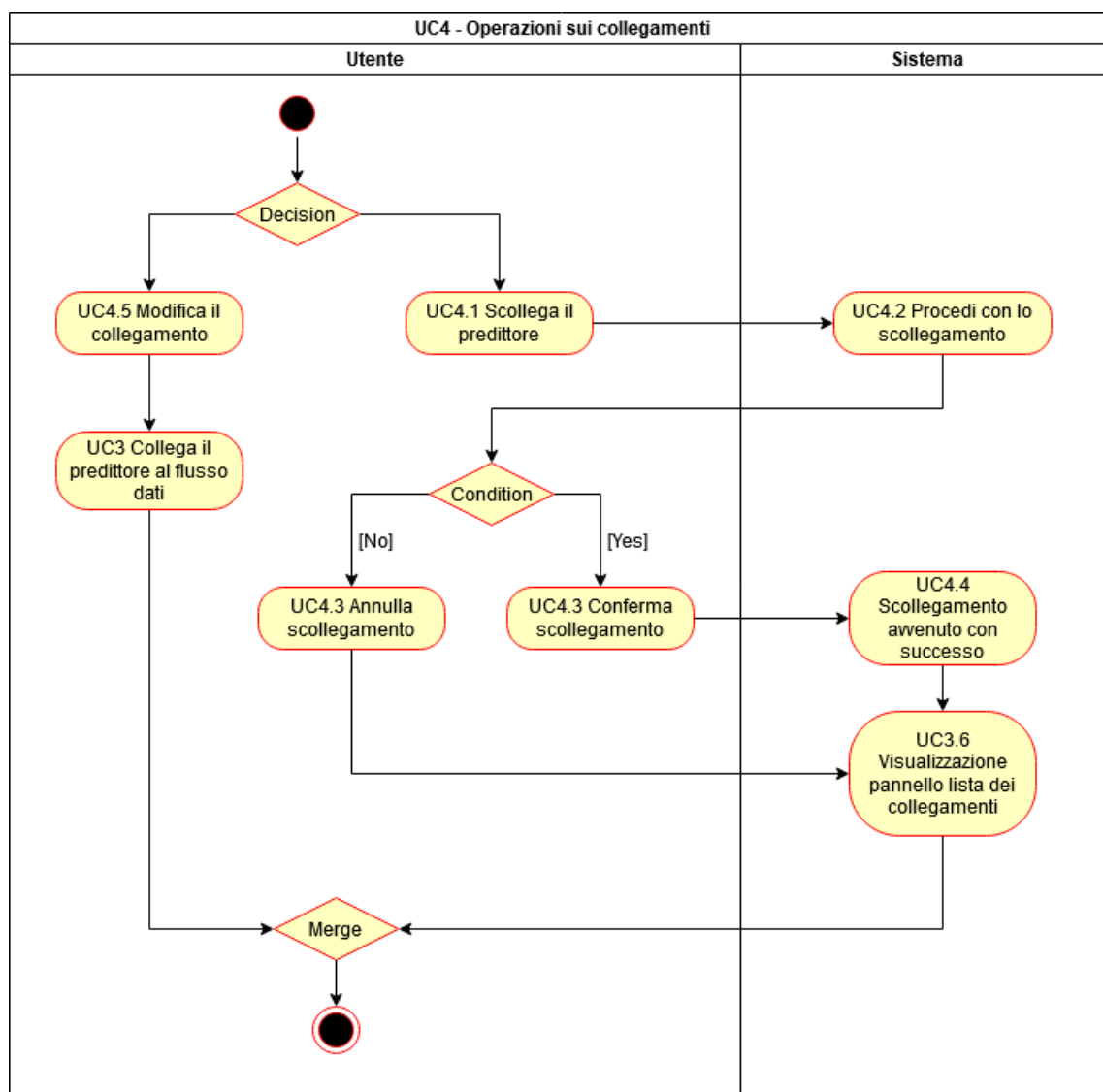


Figura 2.1.4: Diagramma delle attività dello UC4

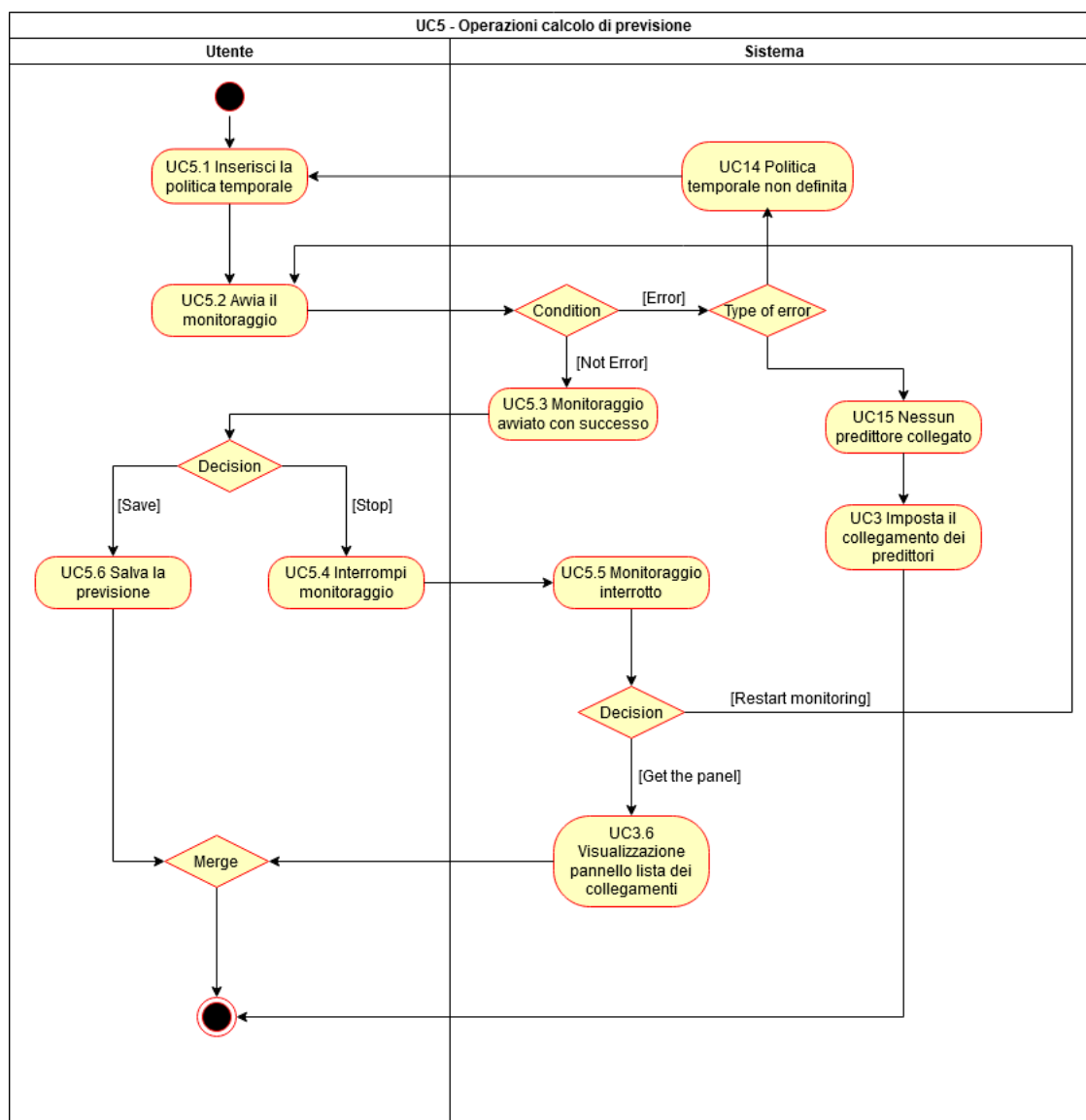


Figura 2.1.5: Diagramma delle attività dello UC5

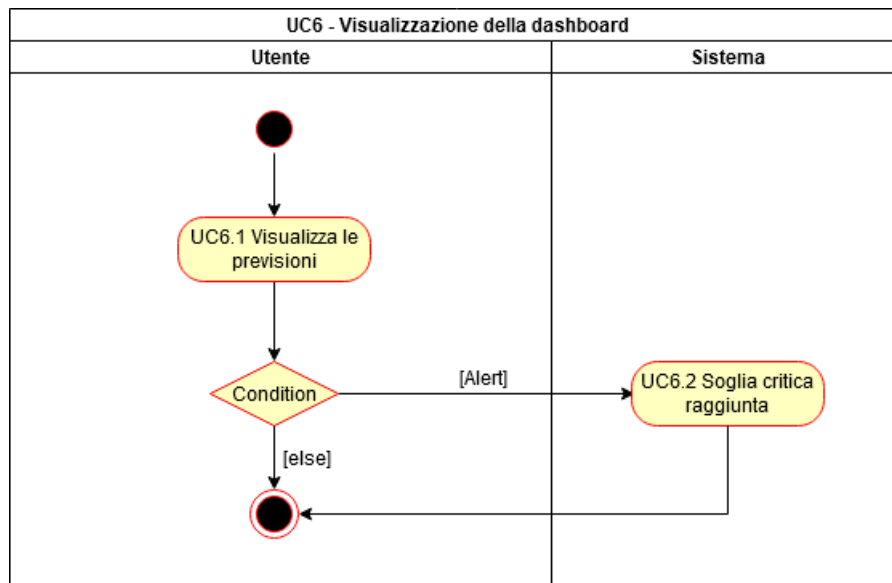


Figura 2.1.6: Diagramma delle attività dello UC6

2.2 Architettura Training Tool

2.2.1 Descrizione

L'implementazione del tool è stata realizzata utilizzando il design pattern MVVM. Il passaggio di dati dalle view al model avviene attraverso la modifica di un campo dati props immesso dal Controller. Attraverso queste props il controller chiama le funzioni corrette quando l'utente interagisce con la vista. La divisione tra business logic e presentation logic è rafforzata da questo utilizzo delle props. Nel controller si trovano un'istanza delle classi SVMtrain e RLtrain in base all'algoritmo scelto, le classi stesse si trovano nel modello e si occupano di rendere polimorfe le funzioni principali, come quelle di addestramento o di recupero del file JSON.

2.2.2 Diagrammi dei package

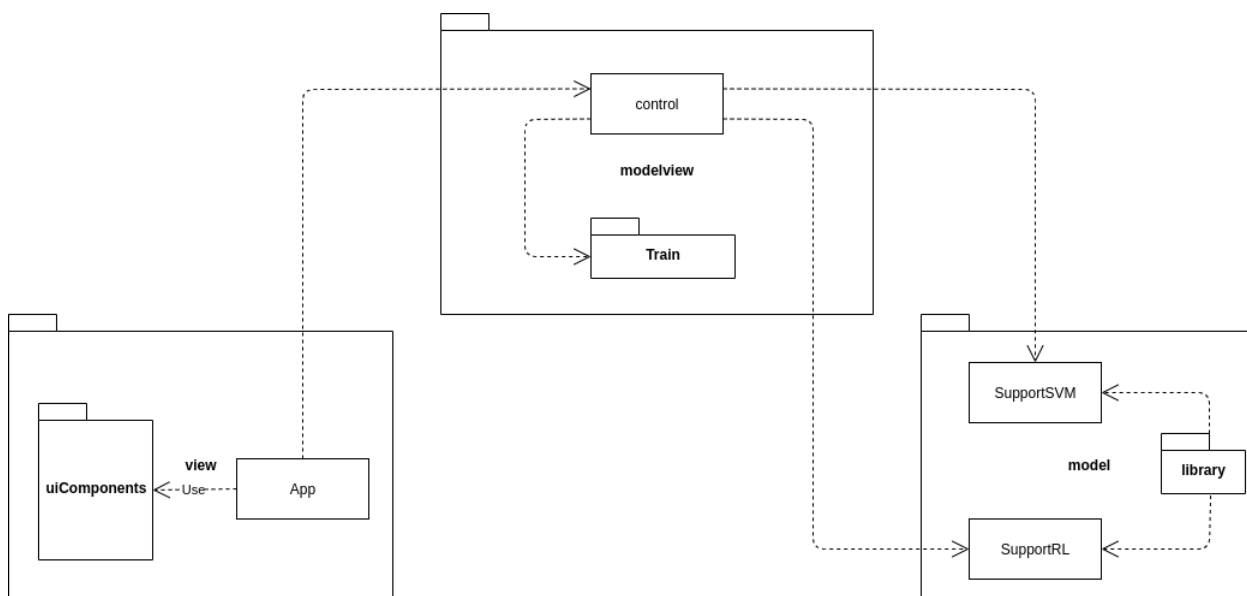


Figura 2.2.1: Diagramma dei package del Training Tool

2.2.3 Diagrammi delle classi

Model

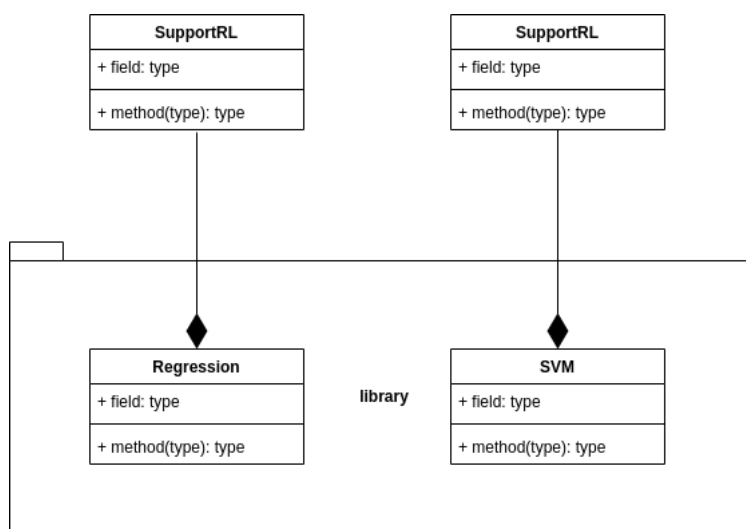


Figura 2.2.2: Diagramma delle classi del Model del Training Tool

View

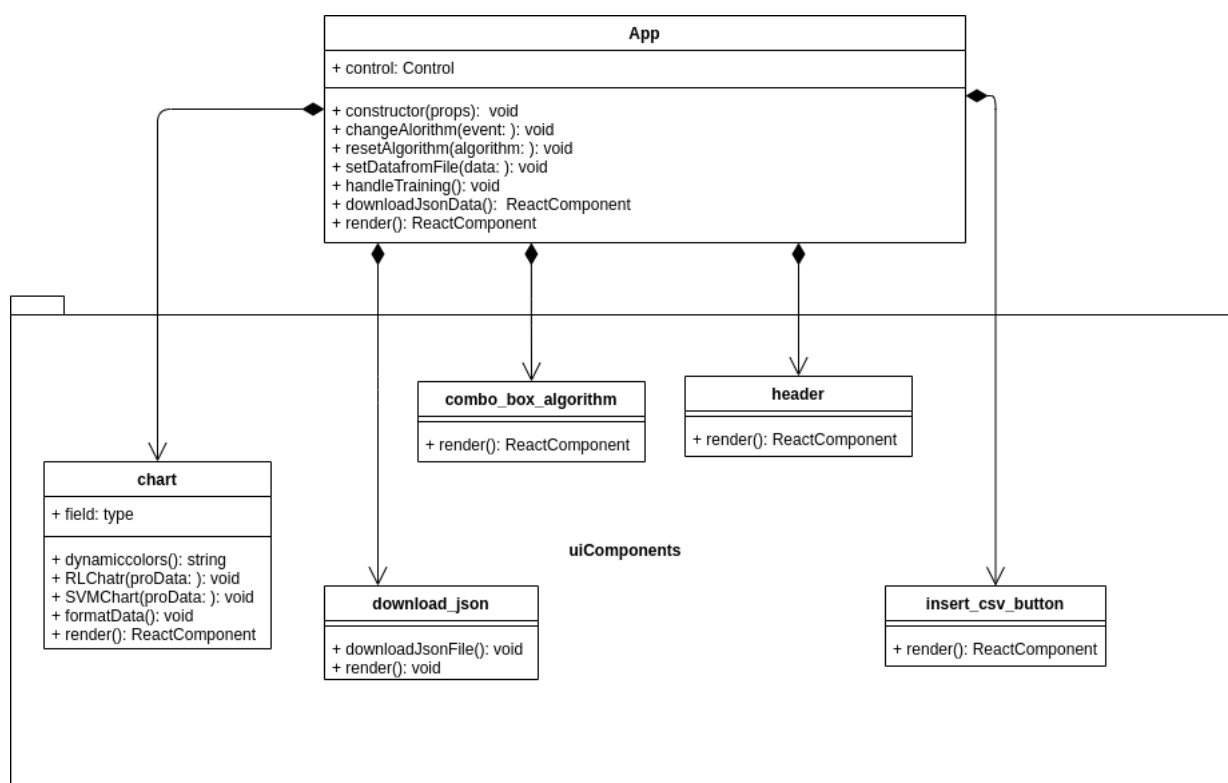


Figura 2.2.3: Diagramma delle classi della View del Training Tool

ViewModel

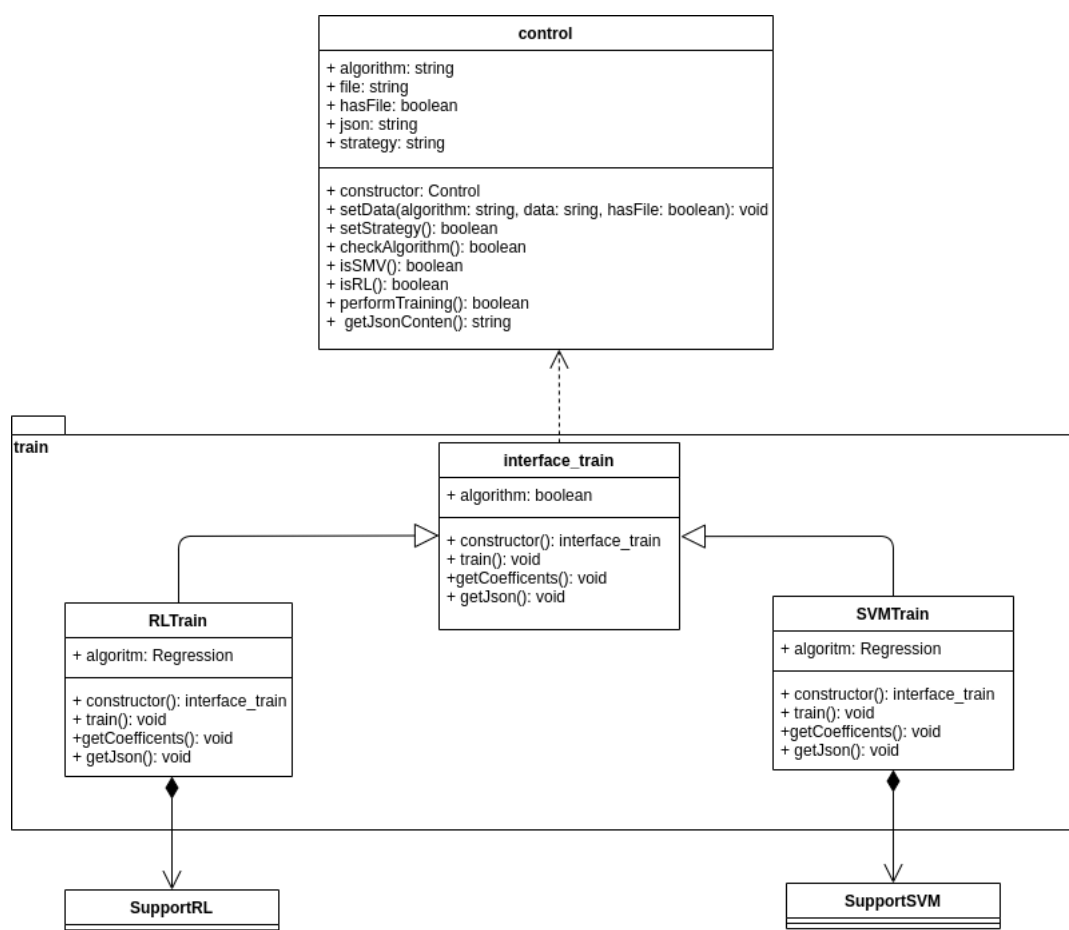


Figura 2.2.4: Diagramma delle classi del ViewModel del Training Tool

2.2.4 Diagrammi di sequenza

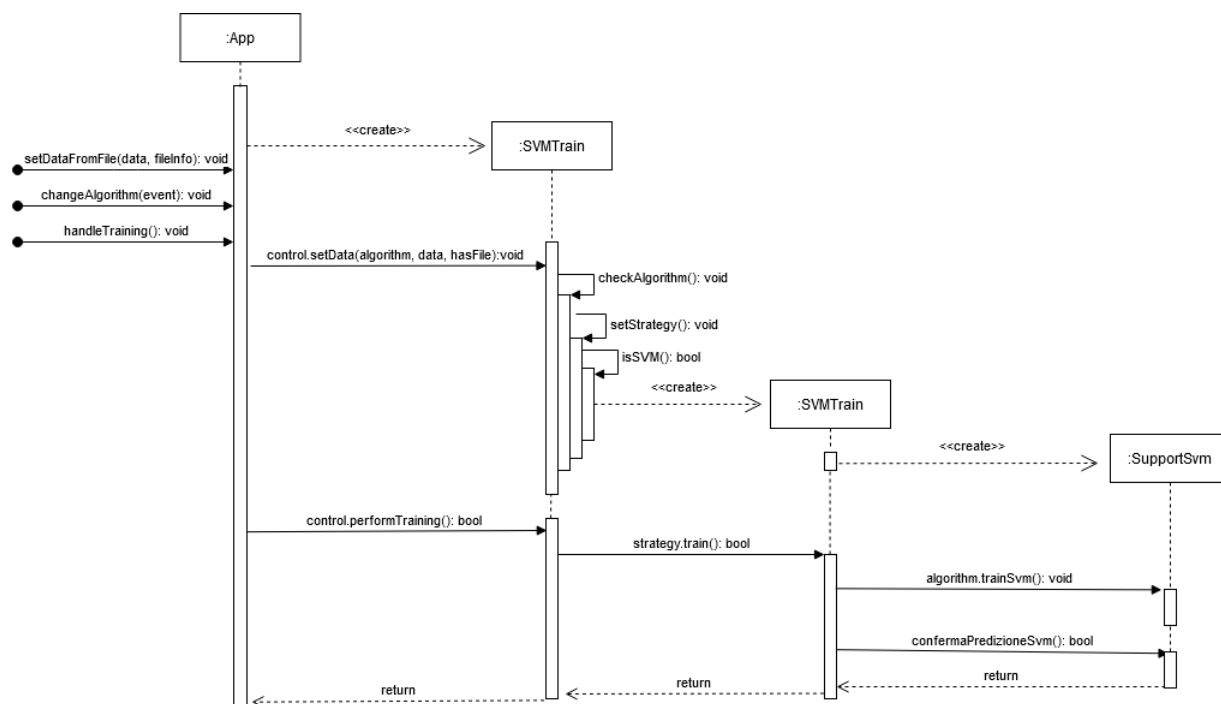


Figura 2.2.5: Diagramma di sequenza del TrainSVM

2.2.5 Design pattern notevoli utilizzati

2.3 Architettura Prediction Plug-in

2.3.1 Descrizione

2.3.2 Diagrammi dei package

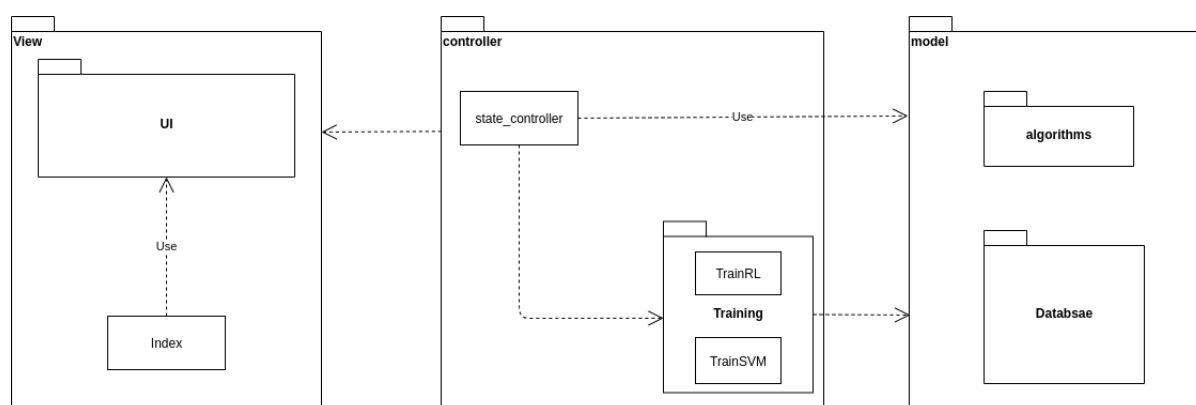


Figura 2.3.1: Diagramma dei package del Prediction Plug-in

2.3.3 Diagrammi delle classi

Model

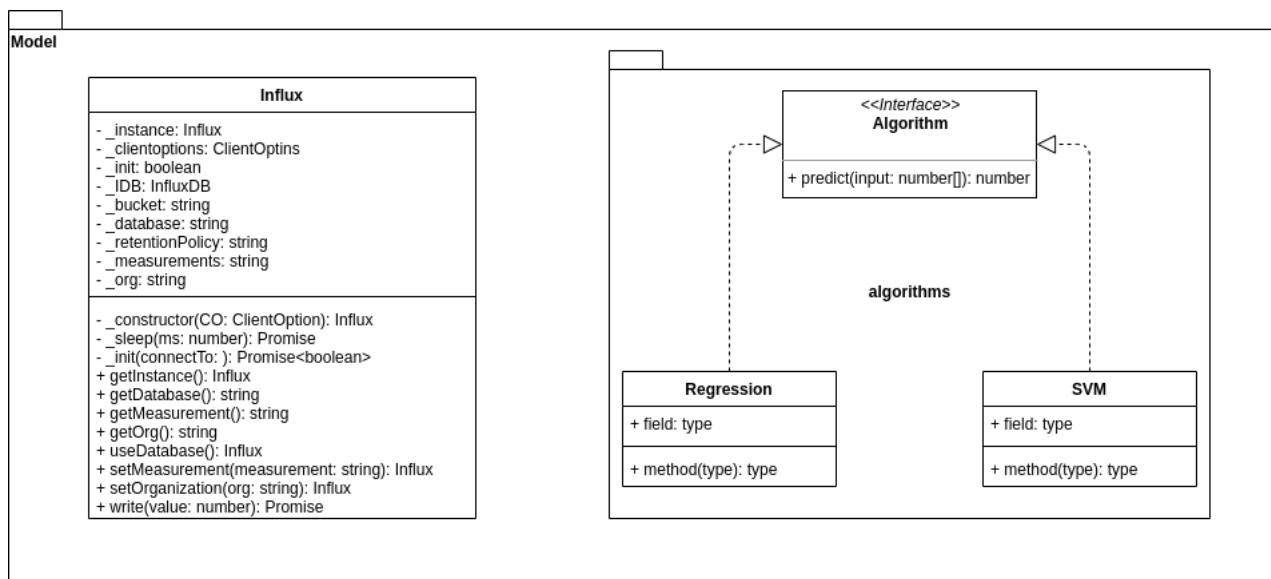


Figura 2.3.2: Diagramma delle classi del Model del Prediction Plug-in

View

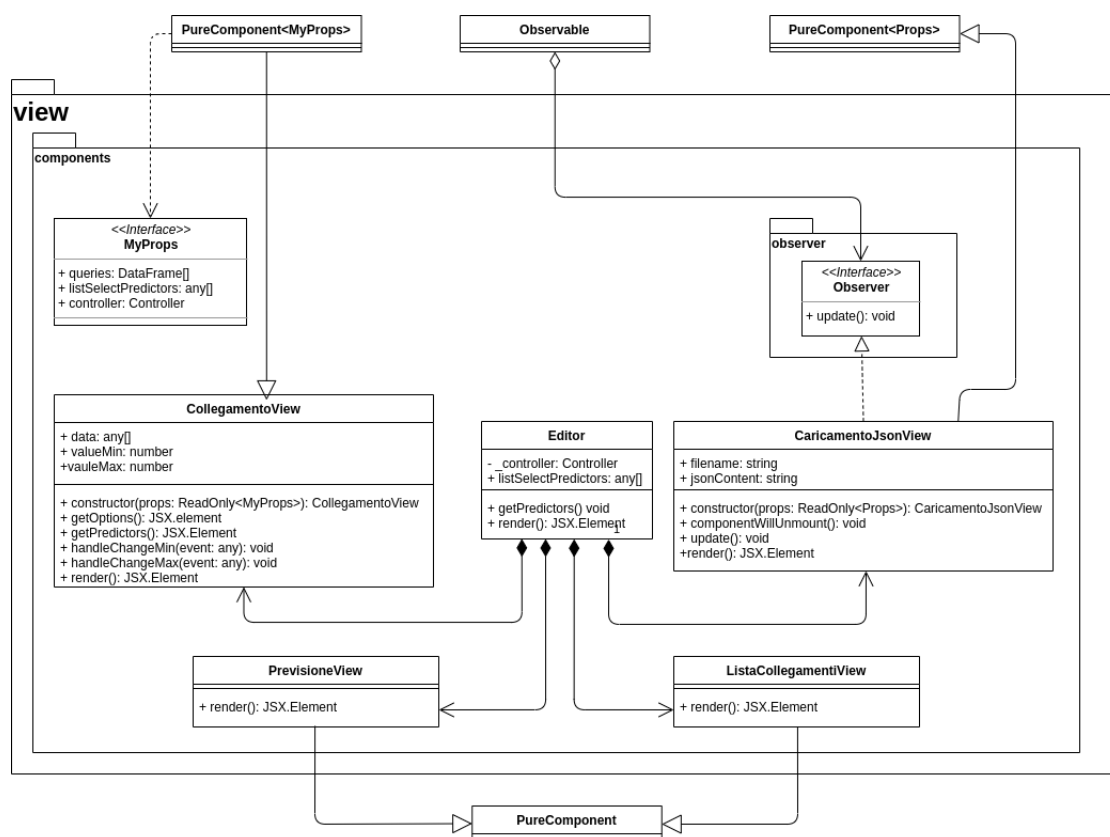


Figura 2.3.3: Diagramma delle classi della View del Prediction Plug-in

Controller

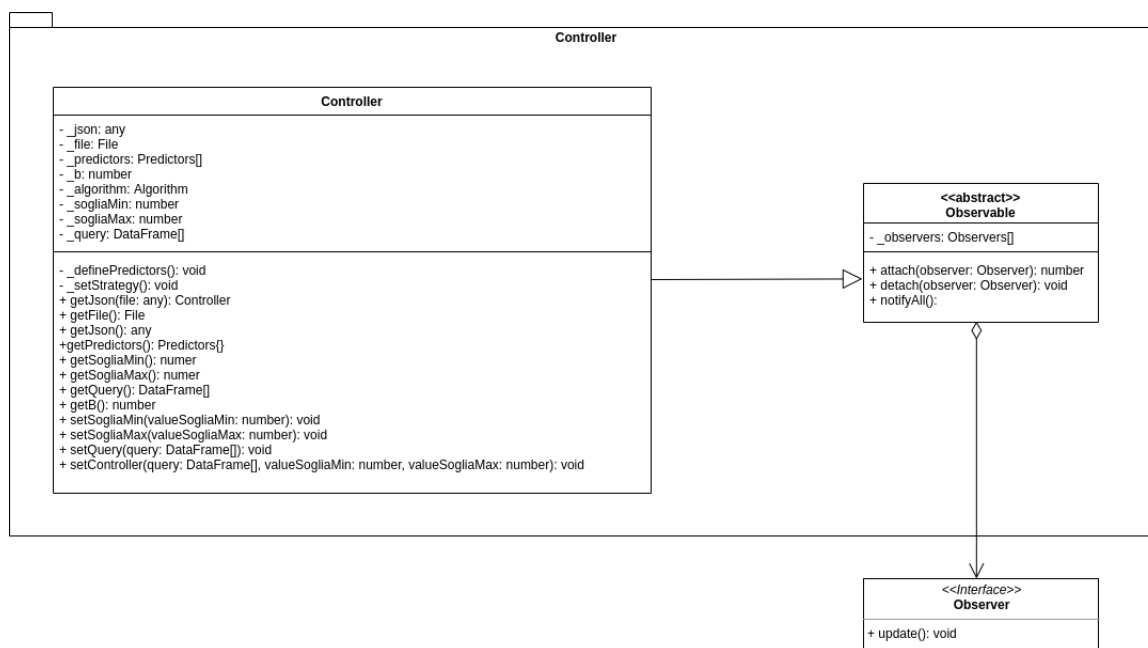


Figura 2.3.4: Diagramma delle classi del Controller del Prediction Plug-in

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
--------	-------

3.2 Grafici del soddisfacimento dei requisiti