TECHNOLOGY BASELINE

AFK

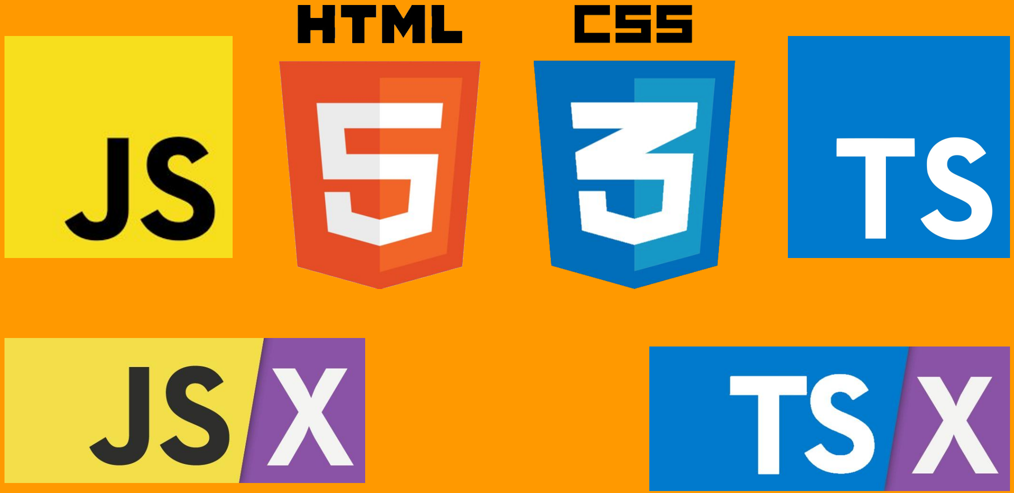
PREDIRE IN GRAFANA

Where your predictions come true

gruppoafk15@gmail.com

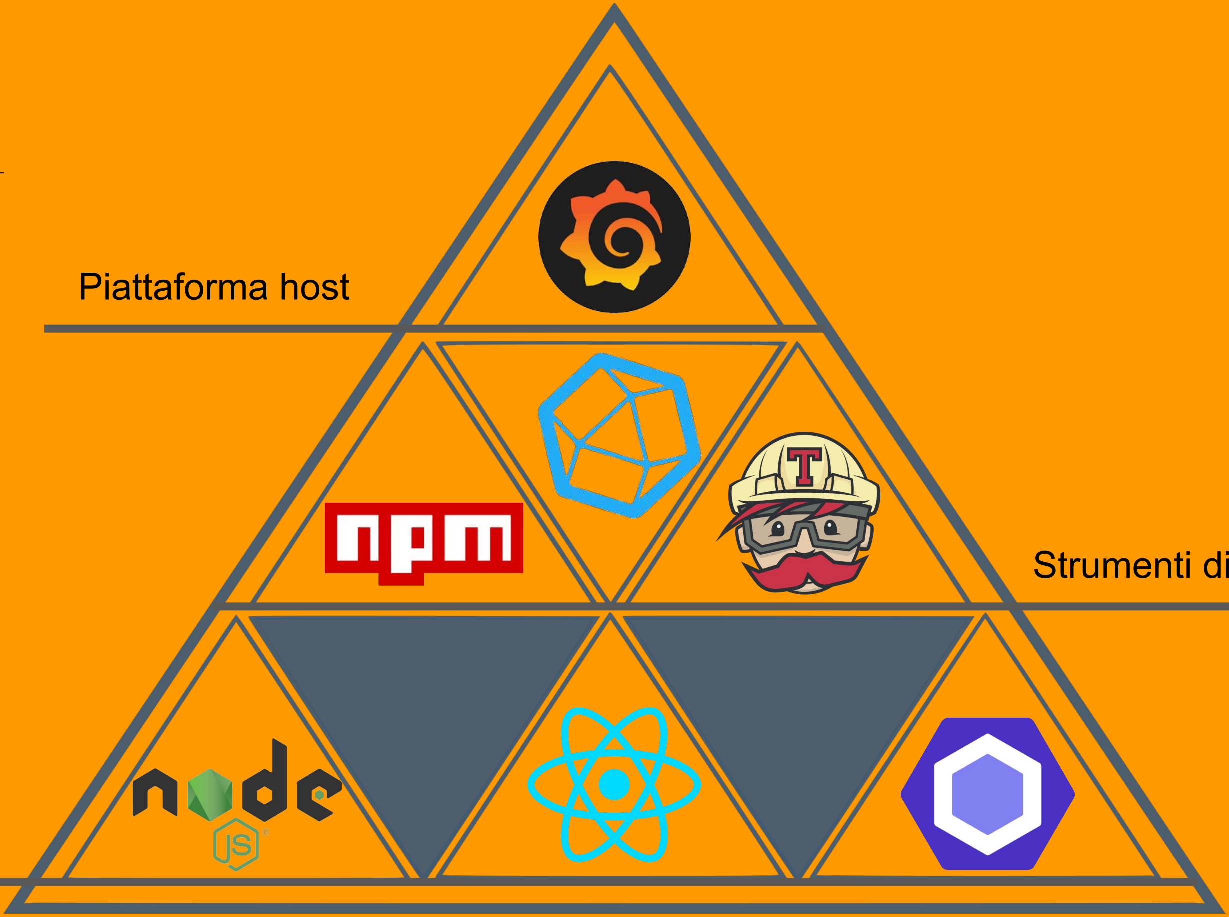
LINGUAGGI UTILIZZATI

TOOL



PLUG-IN

TECNOLOGIE UTILIZZATE

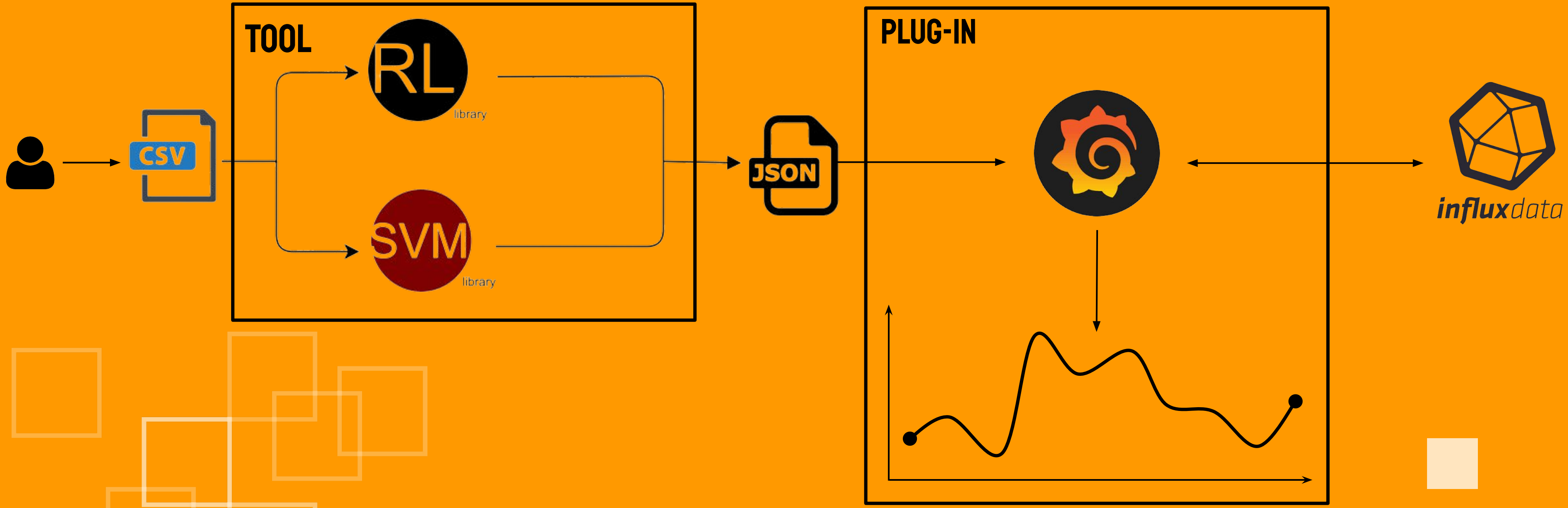


Piattaforma host

Strumenti di supporto

Strumenti di sviluppo

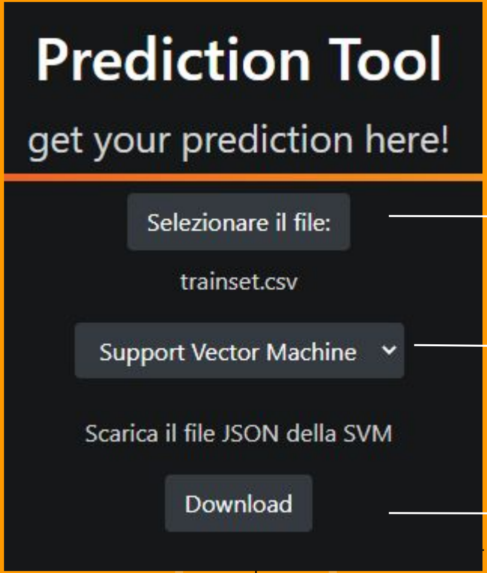
FUNZIONAMENTO GENERALE



CASI D'USO AFFRONTATI

CODICE	IDENTIFICATIVO	CASO D'USO
UC1	UC1.1	Selezione dati di addestramento
	UC1.2	Selezione dell'algoritmo di previsione
	UC1.3	Conferma procedura addestramento
	UC1.4	Salvataggio file JSON
UC2	UC2.1	Selezione del file JSON
UC3	UC3.1	Selezione database
	UC3.2	Selezione del flusso dati
UC9	Estensione UC1.3	Visualizzazione messaggio di errore "File CSV incompatibile"

TOOL & PLUG-IN



Prediction Tool

get your prediction here!

Selezionare il file:

trainset.csv

Support Vector Machine

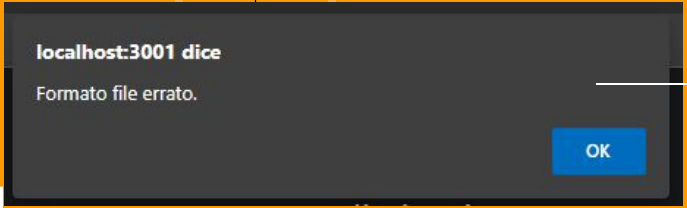
Scarica il file JSON della SVM

Download

input csv

selezione algoritmo

output json

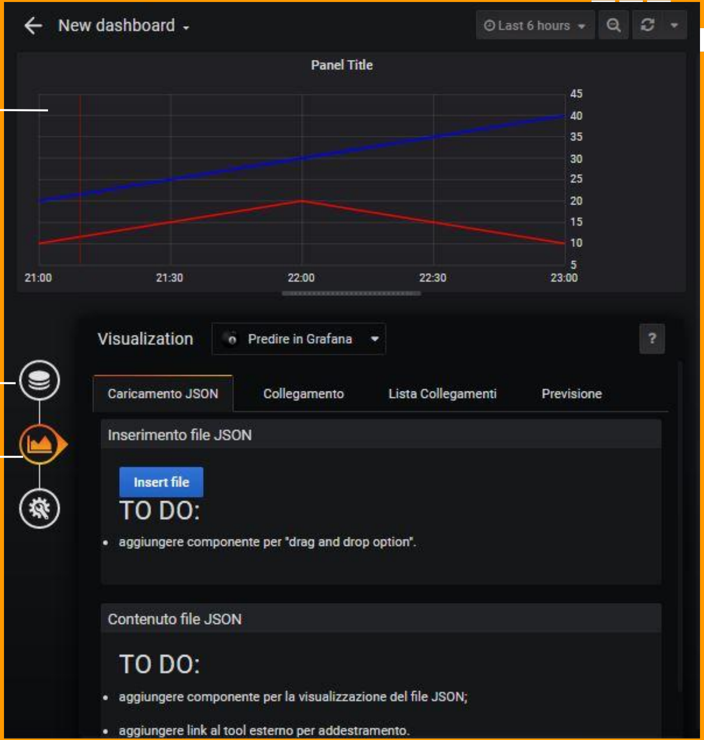


localhost:3001 dice

Formato file errato.

OK

avviso file csv non compatibile



Inserimento file JSON

TO DO:

- aggiungere componente per "drag and drop option".

Contenuto file JSON

TO DO:

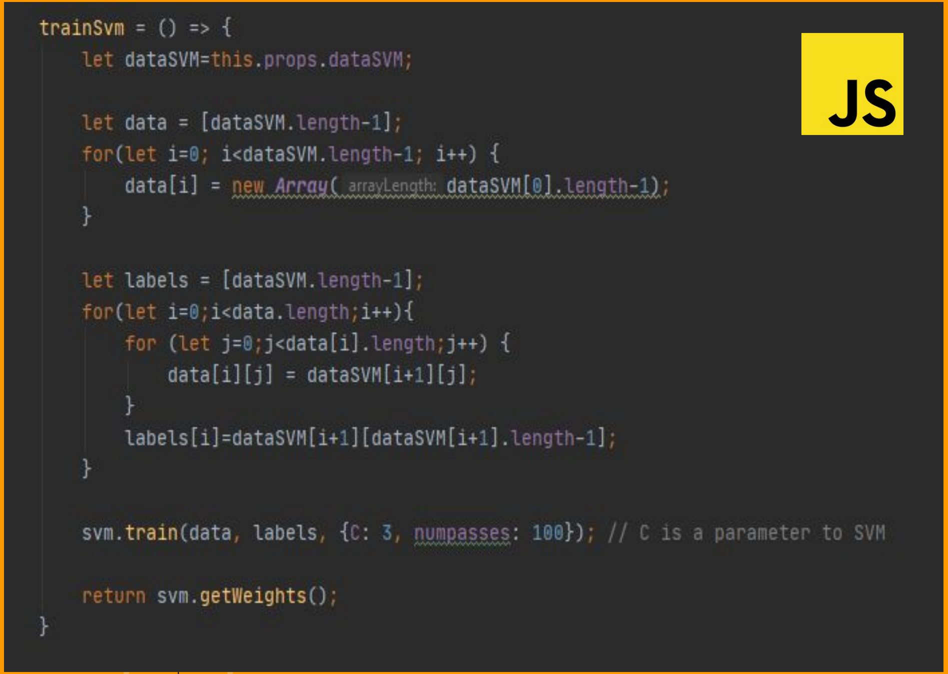
- aggiungere componente per la visualizzazione del file JSON;
- aggiungere link al tool esterno per addestramento.

visualizzazione previsione

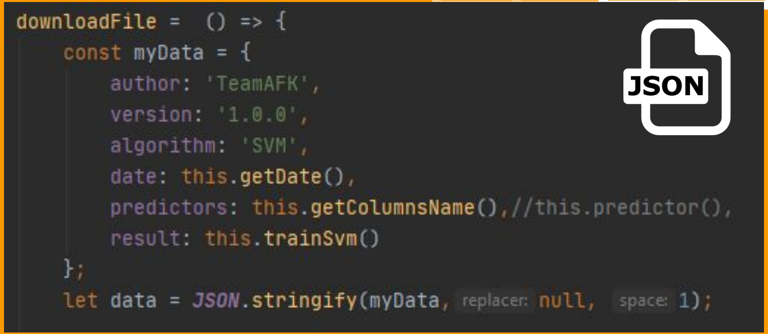
impostazione database e flusso

settaggio predizione

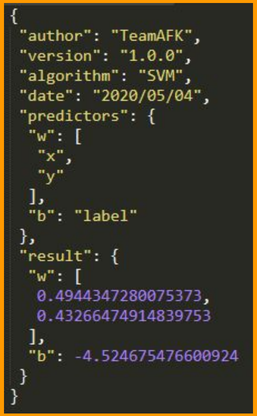
SUPPORT VECTOR MACHINE



```
trainSvm = () => {
    let dataSVM=this.props.dataSVM;

    let data = [dataSVM.length-1];
    for(let i=0; i<dataSVM.length-1; i++) {
        data[i] = new Array( arrayLength: dataSVM[0].length-1);
    }

    let labels = [dataSVM.length-1];
    for(let i=0;i<data.length;i++){
        for (let j=0;j<data[i].length;j++) {
            data[i][j] = dataSVM[i+1][j];
        }
        labels[i]=dataSVM[i+1][dataSVM[i+1].length-1];
    }

    svm.train(data, labels, {C: 3, numpasses: 100}); // C is a parameter to SVM

    return svm.getWeights();
}
```

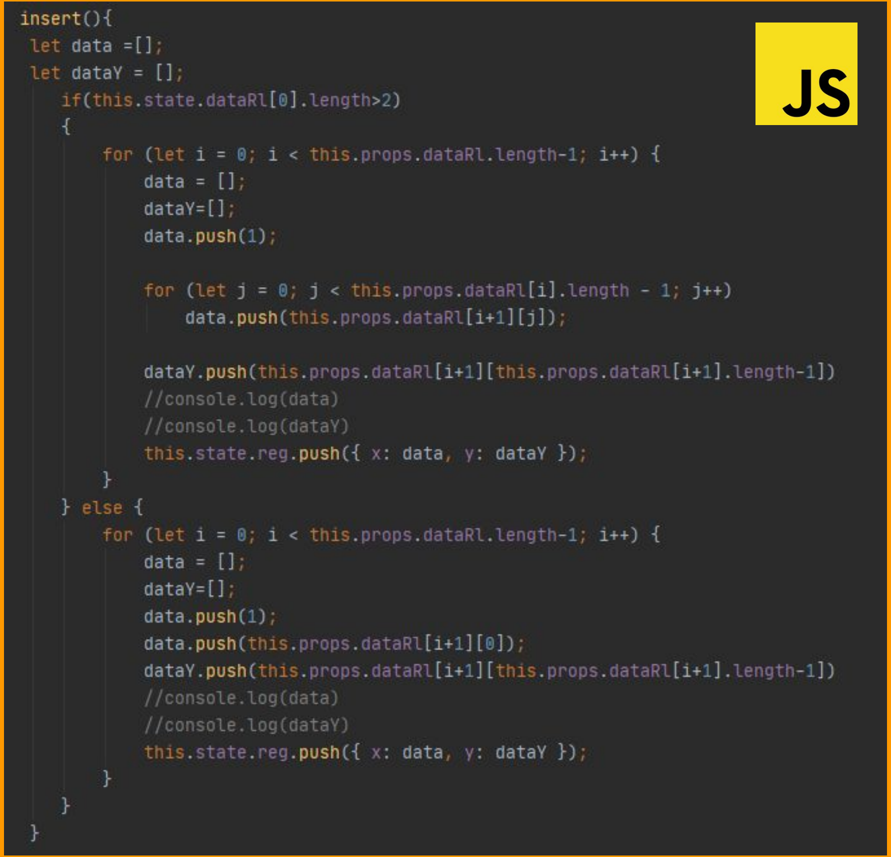


```
downloadFile =  () => {
    const myData = {
        author: 'TeamAFK',
        version: '1.0.0',
        algorithm: 'SVM',
        date: this.getDate(),
        predictors: this.getColumnsName(),//this.predictor(),
        result: this.trainSvm()
    };
    let data = JSON.stringify(myData, replacer: null,  space: 1);
```

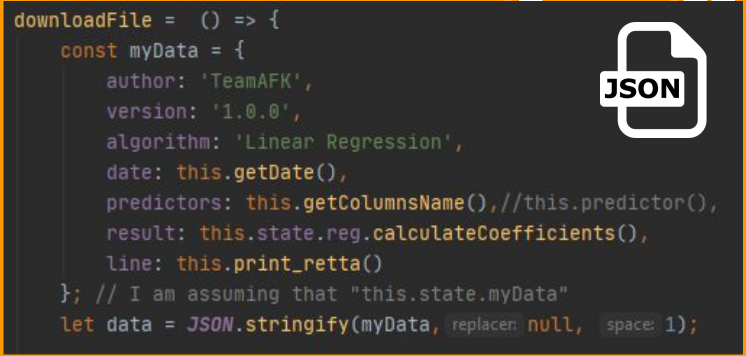


```
{
 "author": "TeamAFK",
 "version": "1.0.0",
 "algorithm": "SVM",
 "date": "2020/05/04",
 "predictors": {
  "w": [
   "x",
   "y"
  ],
  "b": "label"
 },
 "result": {
  "w": [
   0.4944347280075373,
   0.43266474914839753
  ],
  "b": -4.524675476600924
 }
}
```

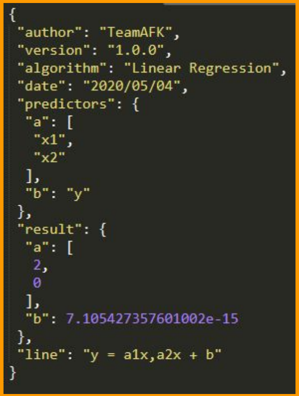
REGRESSIONE LINEARE



```
insert(){
 let data =[];
 let dataY = [];
    if(this.state.dataRl[0].length>2)
    {
        for (let i = 0; i < this.props.dataRl.length-1; i++) {
            data = [];
            dataY=[];
            data.push(1);

            for (let j = 0; j < this.props.dataRl[i].length - 1; j++)
                data.push(this.props.dataRl[i+1][j]);

            dataY.push(this.props.dataRl[i+1][this.props.dataRl[i+1].length-1])
            //console.log(data)
            //console.log(dataY)
            this.state.reg.push({ x: data, y: dataY });
        }
    } else {
        for (let i = 0; i < this.props.dataRl.length-1; i++) {
            data = [];
            dataY=[];
            data.push(1);
            data.push(this.props.dataRl[i+1][0]);
            dataY.push(this.props.dataRl[i+1][this.props.dataRl[i+1].length-1])
            //console.log(data)
            //console.log(dataY)
            this.state.reg.push({ x: data, y: dataY });
        }
    }
 }
```



```
downloadFile =  () => {
    const myData = {
        author: 'TeamAFK',
        version: '1.0.0',
        algorithm: 'Linear Regression',
        date: this.getDate(),
        predictors: this.getColumnsName(),//this.predictor(),
        result: this.state.reg.calculateCoefficients(),
        line: this.print_retta()
    }; // I am assuming that "this.state.myData"
    let data = JSON.stringify(myData, replacer: null, space: 1);
```



```
{
 "author": "TeamAFK",
 "version": "1.0.0",
 "algorithm": "Linear Regression",
 "date": "2020/05/04",
 "predictors": {
  "a": [
   "x1",
   "x2"
  ],
  "b": "y"
 },
 "result": {
  "a": [
   2,
   0
  ],
  "b": 7.105427357601002e-15
 },
 "line": "y = a1x,a2x + b"
}
```