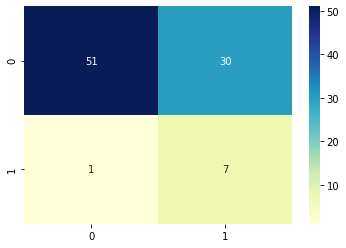
**Texte 1(45)**

**random forest avec échantillon équilibré sans class\_weight avec score: rappel**

{'max\_depth': 5, 'n\_estimators': 100}

score apprentissage:1.0

score test:0.875



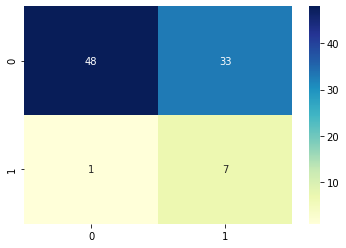
### Cas 2: random forest avec échantillon équilibré sans class\_weight avec score: accurency

{'n\_estimators': 100}

score apprentissage:1.0

score test:0.875

(oublier la max\_depth)

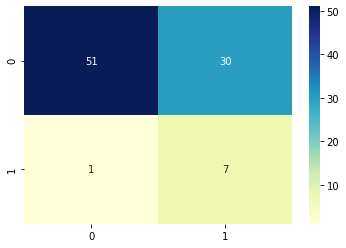


## Random\_forest,accurency,pas boostrap

{'max\_depth': 5, 'max\_features': 'auto', 'n\_estimators': 50}

score apprentissage:1.0

score test:0.875



Random forest, accurency, avec boostrap

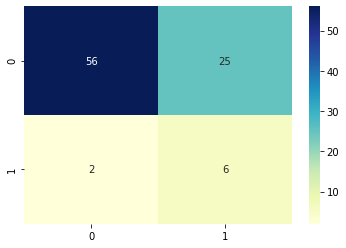
{'max\_depth': 5, 'max\_features': 0.02, 'max\_samples': 0.7, 'n\_estimators': 150}

score apprentissage:1.0

score test:0.75

755.4961769580841

(bug pour le max sample=1)

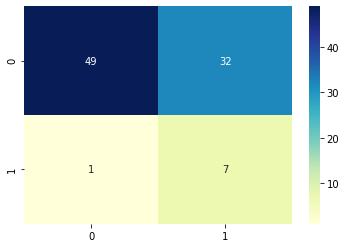


## bagging: KNN,accurency, sans boost

{'base\_estimator\_\_n\_neighbors': 5, 'max\_features': 0.02, 'n\_estimators': 100}

score apprentissage:1.0

score test:0.875

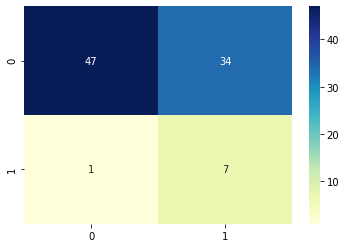


bagging KNN, recall, sans boost

{'base\_estimator\_\_n\_neighbors': 4, 'max\_features': 0.02, 'n\_estimators': 100}

score apprentissage:1.0

score test:0.875



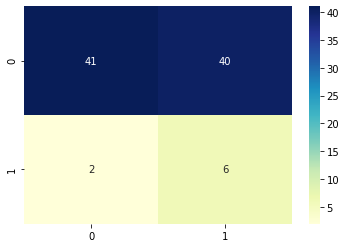
Bagging KNN, accurency, avec boost

{'base\_estimator\_\_n\_neighbors': 10, 'bootstrap\_features': True, 'max\_features': 0.01, 'max\_samples': 1.0, 'n\_estimators': 50}

score apprentissage:1.0

score test:0.75

3441.790701150894

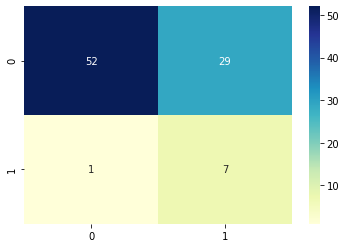


Bagging log reg, recall, sans boost

{'max\_features': 0.2, 'n\_estimators': 20}

score apprentissage:0.9705882352941176

score test:0.875

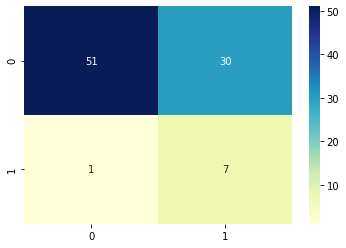


Bagging log reg, accurency, sans boost

{'max\_features': 0.1, 'n\_estimators': 50}

score apprentissage:0.9705882352941176

score test:0.875



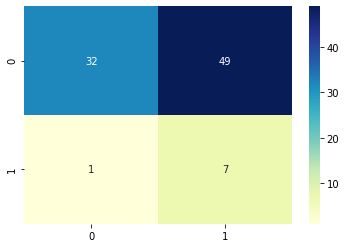
Bagging log reg, recall, avec boost

{'bootstrap\_features': True, 'max\_features': 0.7000000000000001, 'max\_samples': 0.2, 'n\_estimators': 10}

score apprentissage:0.8529411764705882

score test:0.875

637.2448844909668



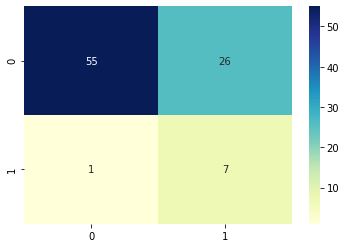
Bagging SVM, recall,avec boost

{'base\_estimator\_\_C': 0.01, 'base\_estimator\_\_degree': 2, 'base\_estimator\_\_kernel': 'linear', 'bootstrap\_features': True, 'max\_features': 0.2, 'max\_samples': 0.5, 'n\_estimators': 20}

score apprentissage:0.7647058823529411

score test:0.875

11561.179428339005



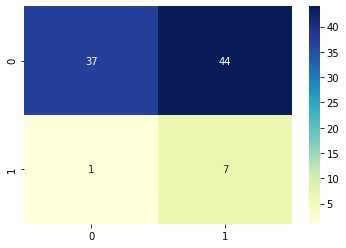
KNN, sans bagging, accurency

{'n\_neighbors': 14}

score apprentissage:1.0

score test:0.875

1.711336612701416



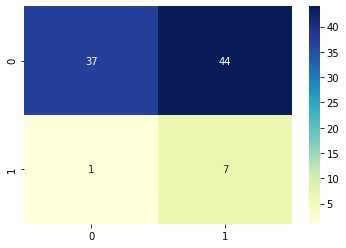
KNN, sans bagging, recall

{'n\_neighbors': 10}

score apprentissage:1.0

score test:0.875

1.2073323726654053



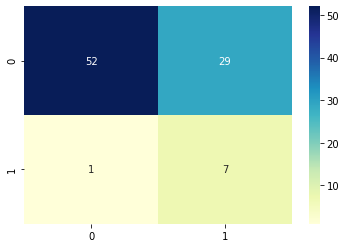
SVM sans bagging,accurency

{'C': 1, 'degree': 3, 'kernel': 'poly'}

score apprentissage:1.0

score test:0.875

4.2396018505096436



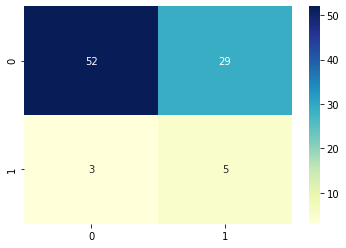
Svm snas bagging, recall

{'C': 100, 'degree': 2, 'kernel': 'poly'}

score apprentissage:1.0

score test:0.625

4.4106574058532715



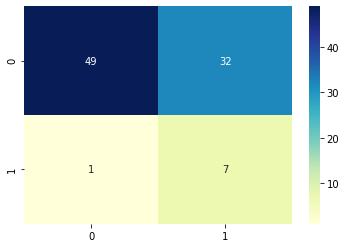
Log reg sans bagging, accurency

{'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}

score apprentissage:0.8823529411764706

score test:0.875

7.846543788909912



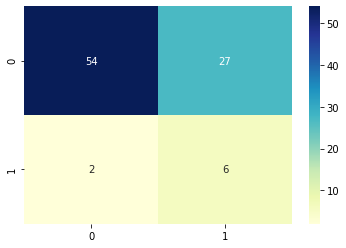
Log reg sans bagging, recall

{'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}

score apprentissage:1.0

score test:0.75

7.840746641159058



Texte 2(1000)

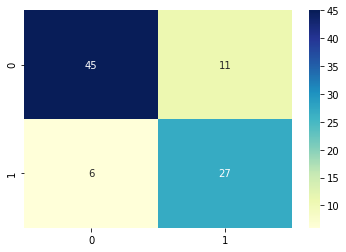
Random forest accurency, boostrap

{'max\_depth': 9, 'max\_features': 'auto', 'max\_samples': 0.999, 'n\_estimators': 200}

score apprentissage:1.0

score test:0.8181818181818182

1524.6006257534027



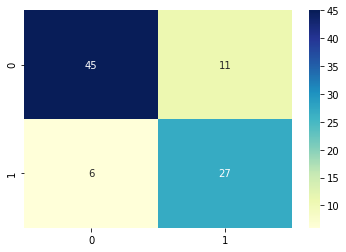
Random forest accurency, sans boost

{'max\_depth': 9, 'n\_estimators': 200}

score apprentissage:1.0

score test:0.8181818181818182

258.7171280384064



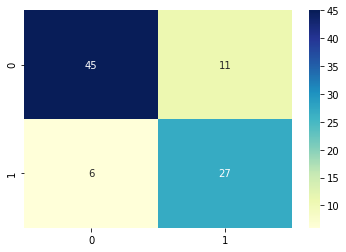
Random forest, recall, sans boost

{'max\_depth': 9, 'n\_estimators': 200}

score apprentissage:1.0

score test:0.8181818181818182

167.50624442100525



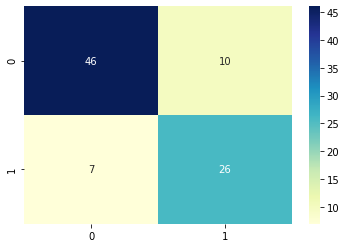
KNN, accurency, sans boost

{'base\_estimator\_\_n\_neighbors': 7, 'bootstrap\_features': False, 'max\_features': 0.1, 'n\_estimators': 100}

score apprentissage:1.0

score test:0.7878787878787878

1344.2646815776825



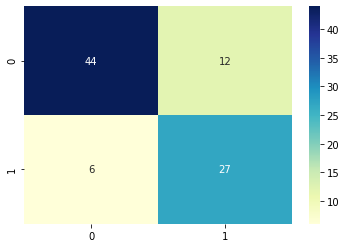
Knn, accurency, sans boost , avec plus grand echantillon de max feat

{'base\_estimator\_\_n\_neighbors': 4, 'max\_features': 0.7, 'n\_estimators': 80}

score apprentissage:1.0

score test:0.8181818181818182

3400.5783224105835



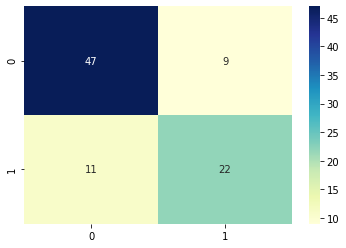
Log reg, boostrap, recall

{'bootstrap\_features': True, 'max\_features': 0.6, 'max\_samples': 0.999, 'n\_estimators': 10}

score apprentissage:0.946969696969697

score test:0.6666666666666666

350.2186596393585



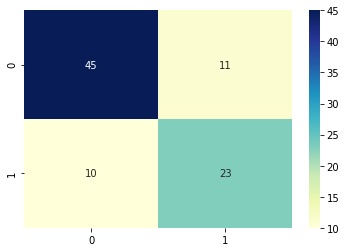
Log reg sans boost,accurency

{'max\_features': 0.30000000000000004, 'n\_estimators': 20}

score apprentissage:0.9696969696969697

score test:0.696969696969697

52.02302813529968



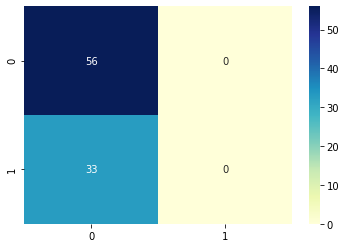
SVM,recall, avec boost

{'base\_estimator\_\_C': 0.01, 'base\_estimator\_\_degree': 2, 'base\_estimator\_\_kernel': 'poly', 'bootstrap\_features': True, 'max\_features': 0.8, 'max\_samples': 0.7, 'n\_estimators': 10}

score apprentissage:0.0

score test:0.0

15549.222067832947



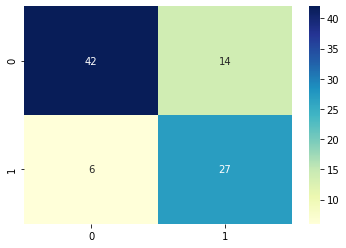
KNN sans bagging

{'n\_neighbors': 3}

score apprentissage:1.0

score test:0.8181818181818182

5.847988128662109



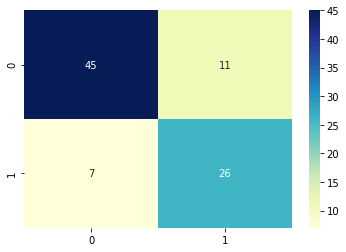
SVm sans bagging,accurency

{'C': 1, 'degree': 2, 'kernel': 'rbf'}

score apprentissage:0.9318181818181818

score test:0.7878787878787878

43.67389965057373



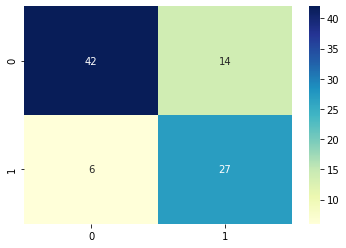
Svm,sans bagging, recall

{'C': 100, 'degree': 2, 'kernel': 'rbf'}

score apprentissage:1.0

score test:0.8181818181818182

41.307170391082764



Bon score en changeant un peu le seuil

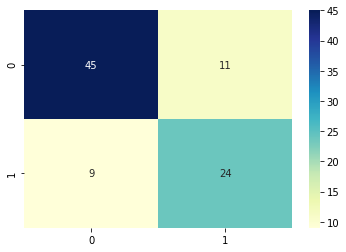
Log reg sans bagging,accurency

{'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}

score apprentissage:0.8257575757575758

score test:0.7272727272727273

25.426368951797485



Log reg sans bagging recall

{'C': 1, 'penalty': 'l1', 'solver': 'saga'}

score apprentissage:0.9848484848484849

score test:0.7272727272727273

26.286107540130615

