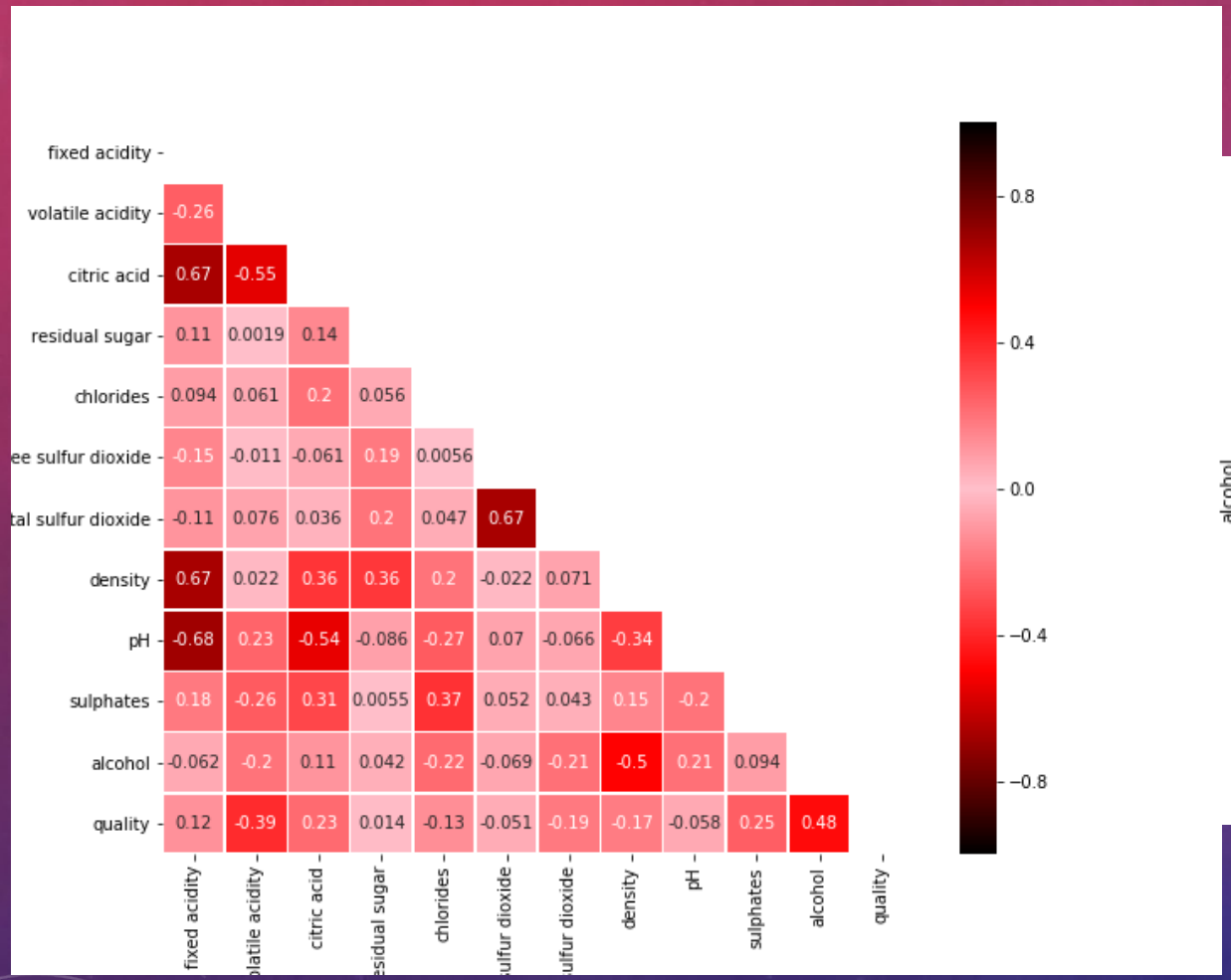
PRÉDIRE LA QUALITÉ DU VIN

APPRENTISSAGE AUTOMATIQUE SUPERVISÉ

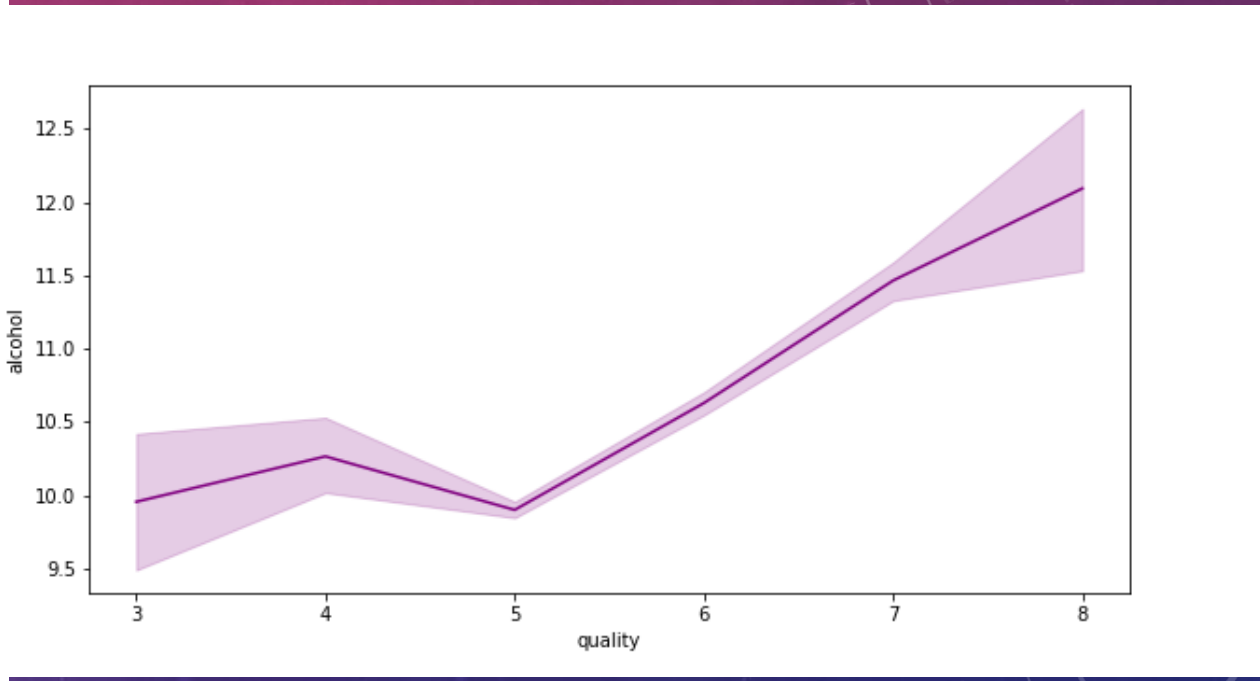
ORIGINE ET PRÉSENTATION DES DONNÉES

- Provenance du dataset : UCI Machine Learning
- Données : taux des composants chimiques (acide tartarique, acide acetique, acide citrique, sucre résiduel, chlorides, dioxyde de soufre, sulfites, alcool, score de qualité)
- Cible : qualité (score entre 0 et 10)
- Etat du dataset : 1599 valeurs, 12 features, aucune valeur manquante, format numérique
- Pas de feature engineering nécessaire

Matrice de corrélation des features



Corrélation alcool et qualité



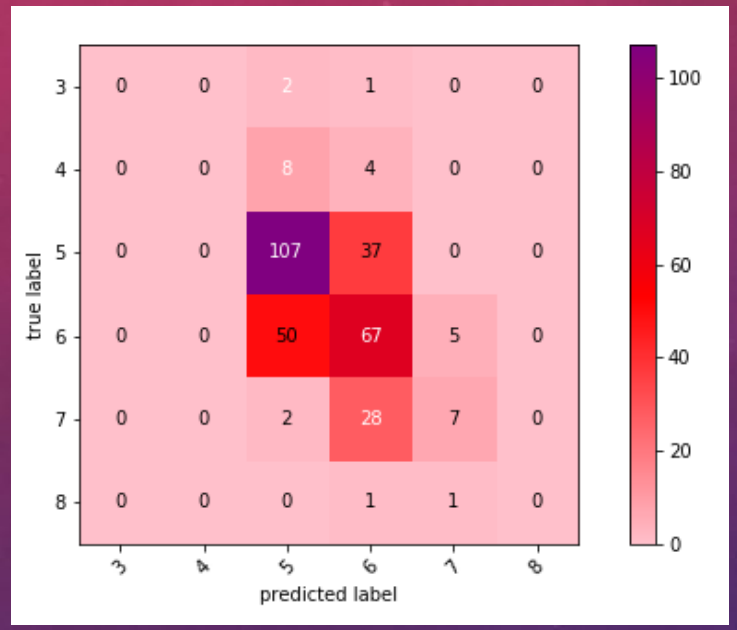
OBJECTIF 1 : PRÉDICTION DE LA QUALITÉ DES VINS
CLASSIFICATION MULTI-CLASSE

- Préparation de la donnée : scalling avec StandardScaler
- Découpage en X_train (shape: 959,11), X_dev (320,11) et X_test (320,11)
- Modèles choisis : Linear SVC, Random Forest, XGBoost
- Métriques d'évaluation : Accuracy score + accuracy score après cross-validation, matrice de confusion, F1 score.

LINEAR SVC

Paramètre obligatoire pour classification multi-classe

```
linearsvc = LinearSVC(dual=False)
linearsvc.fit(X_train,y_train)
pred_svc=linearsvc.predict(X_dev)
```



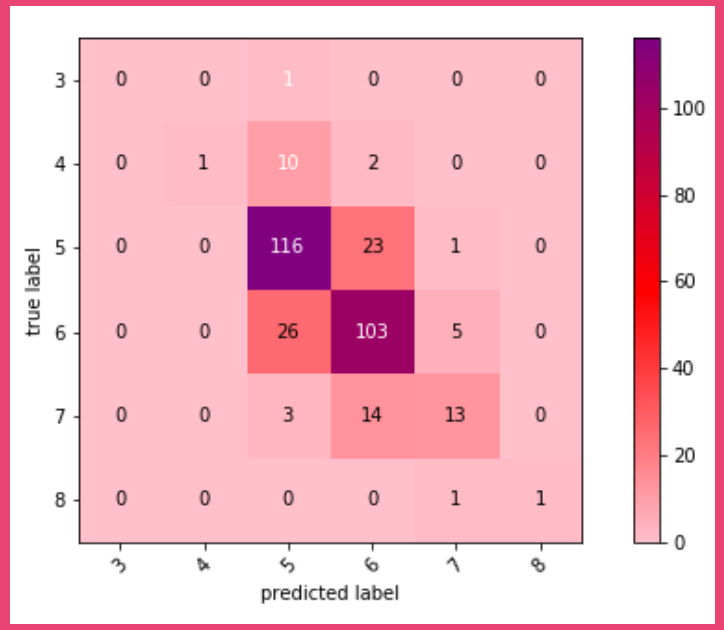
Accuracy score : 0.566

Accuracy score (cross-val) : 0.587

F1 score : 0.566

RANDOM FOREST

```
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
pred_rfc = rfc.predict(X_dev)
```



Accuracy score : 0.682

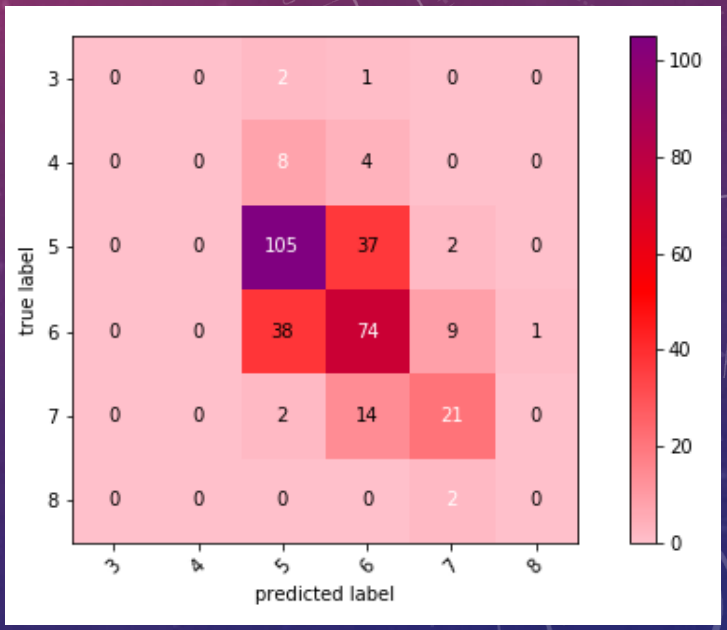
Accuracy score (cross-val) : 0.674

F1 score : 0.681

XGBOOST

Paramètre obligatoire pour classification multi-classe

```
xgboost = XGBClassifier(objective='multi:softmax',
        num_class=10, n_jobs=-1,booster="gbtree",
        tree_method = "hist",grow_policy = "lossguide")
xgboost.fit(X_train, y_train)
pred_xgboost = xgboost.predict(X_dev)
```

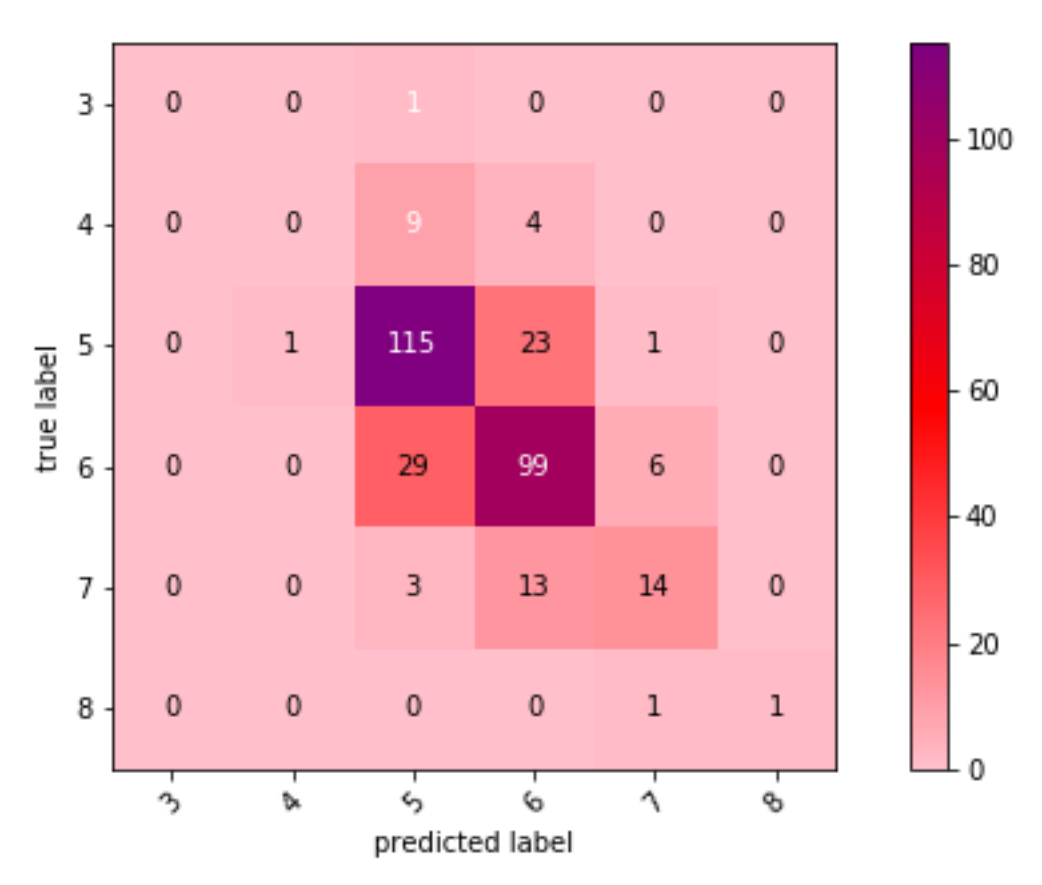


Accuracy score : 0.625

Accuracy score (cross-val) : 0.641

F1 score : 0.625

RANDOM FOREST
PRÉDICTION SUR X_TRAIN COMPLET



Accuracy score : 0.728

Accuracy score
(cross-val) : 0.676

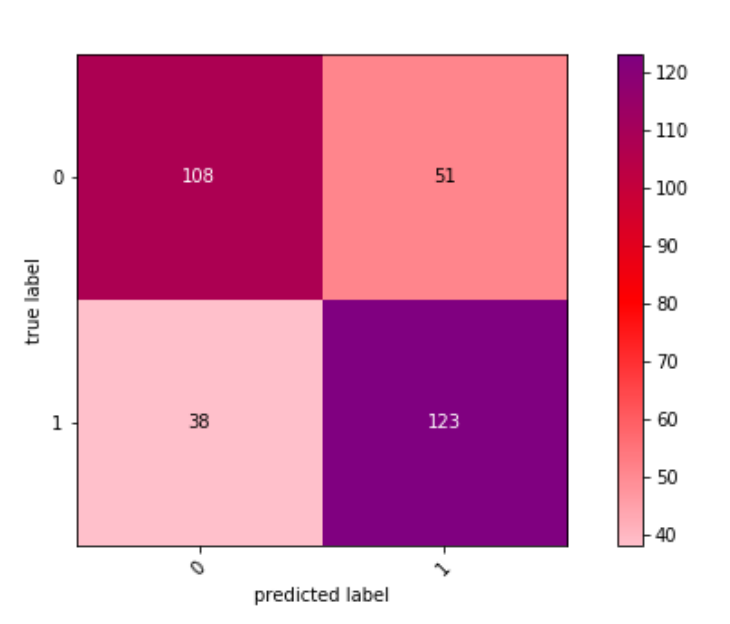
F1 score : 0.722

OBJECTIF 2 : PRÉDICTION BINAIRE DE LA QUALITÉ DES VINS CLASSIFICATION BINAIRE

- Préparation de la donnée : scalling avec StandardScaler, **hot encoding de la feature qualité (<=5=0, >5=1)**
- Découpage en X_train (shape: 959,11), X_dev (320,11) et X_test (320,11)
- Modèles choisis : Linear SVC, Random Forest, XGBoost
- Métriques d'évaluation : Accuracy score + accuracy score après cross-validation, matrice de confusion, F1 score.

LINEAR SVC

```
linearsvc = LinearSVC(dual=True)
linearsvc.fit(X_train,y_train)
pred_svc=linearsvc.predict(X_dev)
```



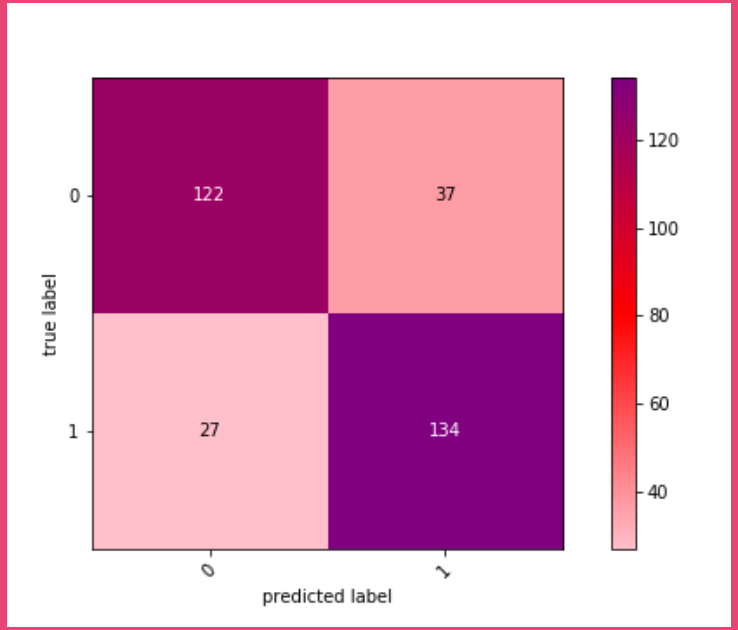
Accuracy score : 0.721

Accuracy score (cross-val) : 0.743

F1 score : 0.722

RANDOM FOREST

```
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
pred_rfc = rfc.predict(X_dev)
```



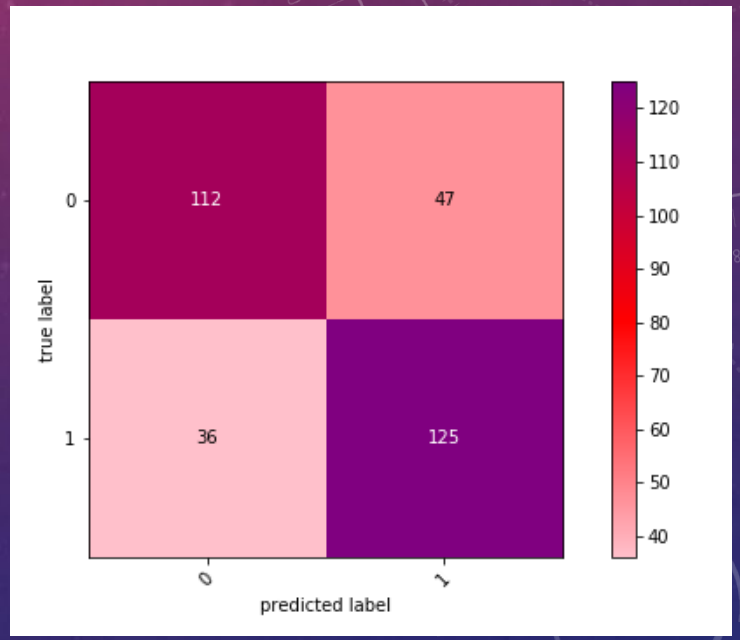
Accuracy score : 0.8

Accuracy score (cross-val) : 0.796

F1 score : 0.8

XGBOOST

```
xgboost = XGBClassifier(objective='reg:logistic',
        n_jobs=-1,booster="gbtree",tree_method = "hist")
xgboost.fit(X_train, y_train)
pred_xgboost = xgboost.predict(X_dev)
```

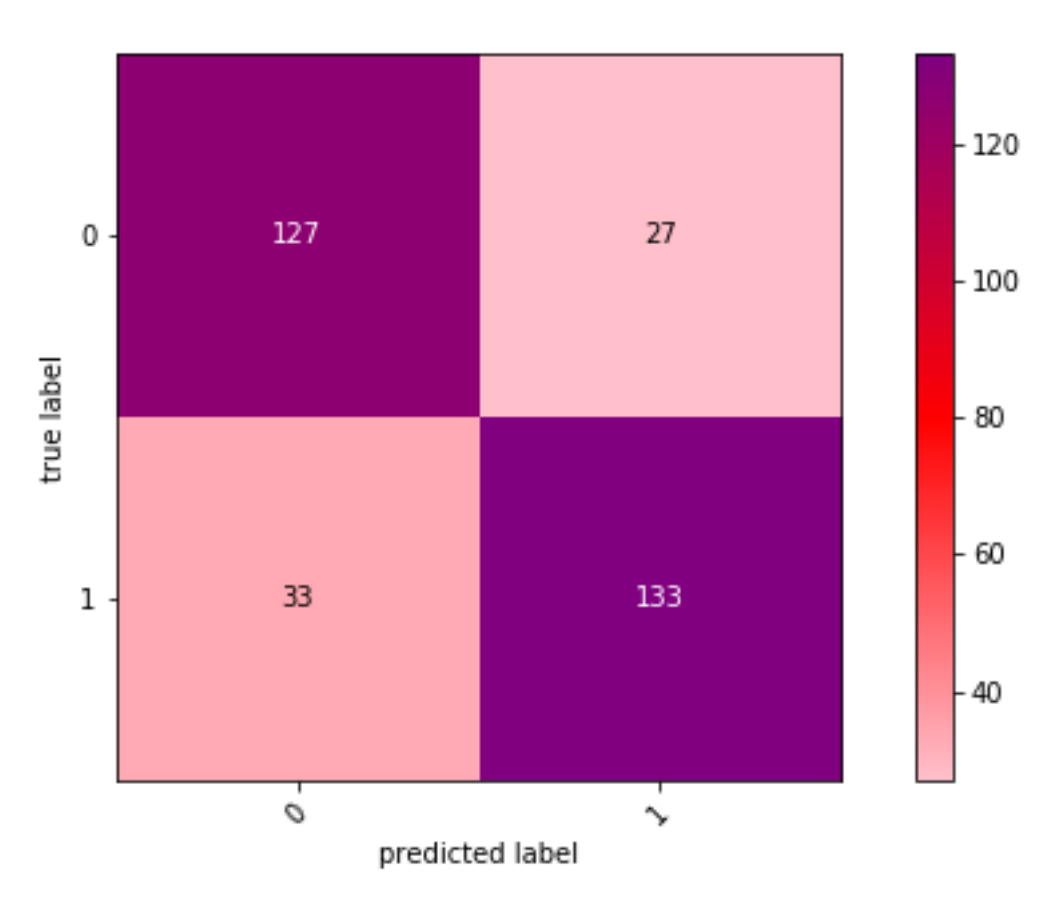


Accuracy score : 0.74

Accuracy score (cross-val) : 0.774

F1 score : 0.741

RANDOM FOREST
PRÉDICTION SUR X_TRAIN COMPLET



Accuracy score : 0.813

Accuracy score
(cross-val) : 0.797

F1 score : 0.812

MERCI!