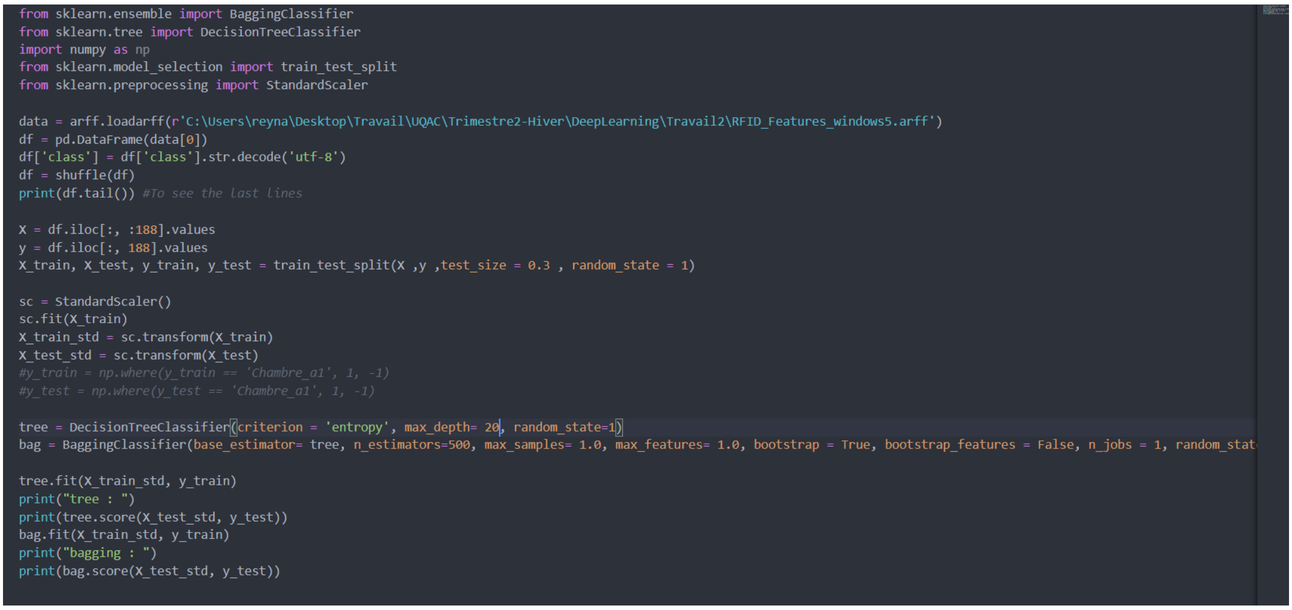
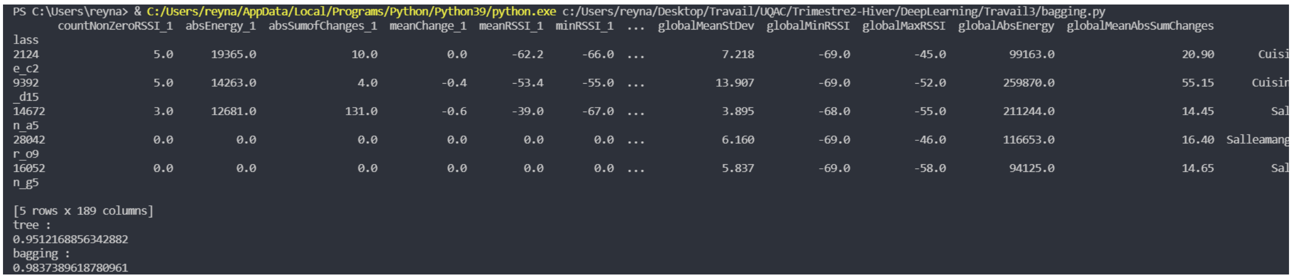
Bagging

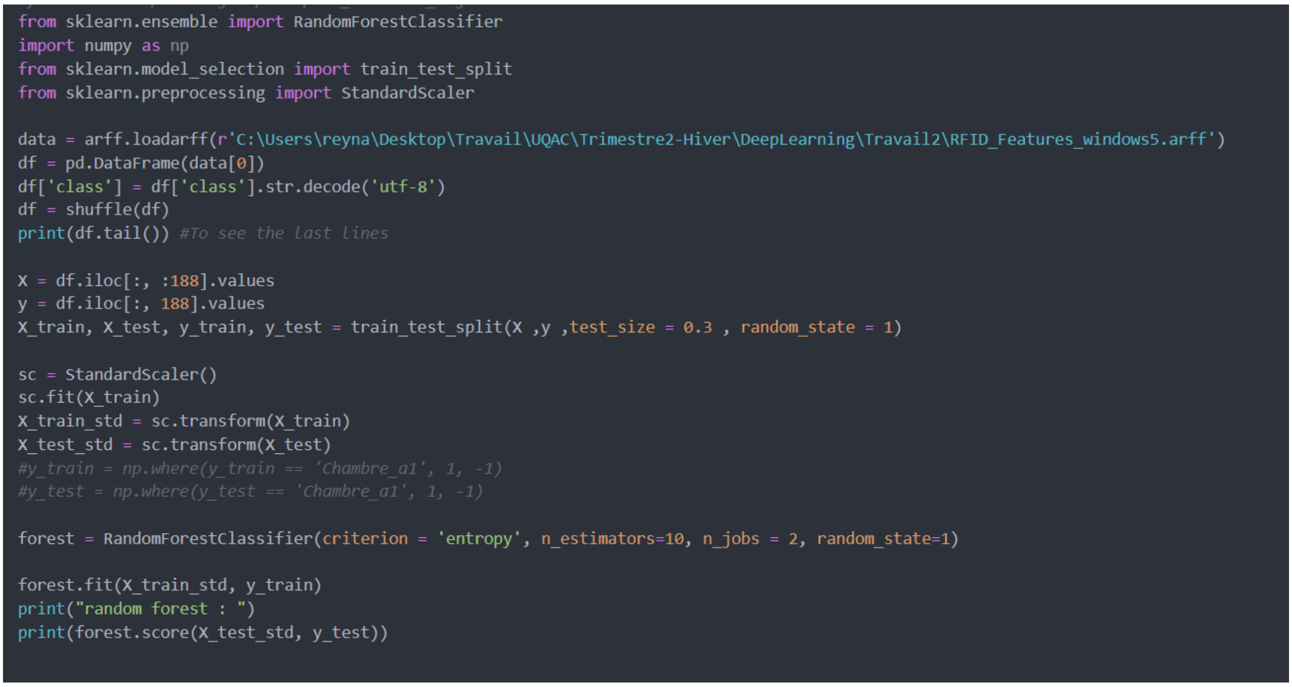


```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

data = arff.loadarff(r'C:\Users\reyna\Desktop\Travail\UQAC\Trimestre2-Hiver\DeepLearning\Travail2\RFID_Features_windows5.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df = shuffle(df)
print(df.tail()) #To see the last lines

X = df.iloc[:, :188].values
y = df.iloc[:, 188].values
X_train, X_test, y_train, y_test = train_test_split(X ,y ,test_size = 0.3 , random_state = 1)

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
#y_train = np.where(y_train == 'Chambre_a1', 1, -1)
#y_test = np.where(y_test == 'Chambre_a1', 1, -1)

tree = DecisionTreeClassifier(criterion = 'entropy', max_depth= 20, random_state=1)
bag = BaggingClassifier(base_estimator= tree, n_estimators=500, max_samples= 1.0, max_features= 1.0, bootstrap = True, bootstrap_features = False, n_jobs = 1, random_stat

tree.fit(X_train_std, y_train)
print("tree : ")
print(tree.score(X_test_std, y_test))
bag.fit(X_train_std, y_train)
print("bagging : ")
print(bag.score(X_test_std, y_test))
```

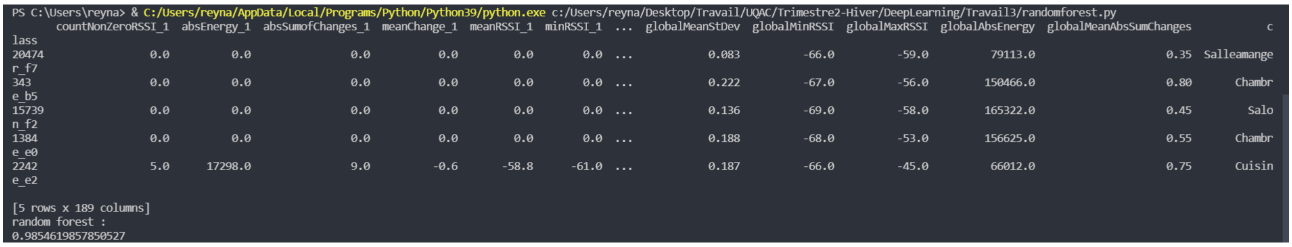


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail3/bagging.py
       countNonZeroRSSI_1  absEnergy_1  absSumofChanges_1  meanChange_1  meanRSSI_1  minRSSI_1  ...  globalMeanStDev  globalMinRSSI  globalMaxRSSI  globalAbsEnergy  globalMeanAbsSumChanges
lass
2124                  5.0      19365.0               10.0           0.0       -62.2      -66.0  ...            7.218          -69.0          -45.0          99163.0                    20.90       Cuisi
e_c2
9392                  5.0      14263.0                4.0          -0.4       -53.4      -55.0  ...           13.907          -69.0          -52.0         259870.0                    55.15      Cuisin
_d15
14672                 3.0      12681.0              131.0          -0.6       -39.0      -67.0  ...            3.895          -68.0          -55.0         211244.0                    14.45         Sal
n_a5
28042                 0.0          0.0                0.0           0.0         0.0        0.0  ...            6.160          -69.0          -46.0         116653.0                    16.40  Salleamang
r_o9
16052                 0.0          0.0                0.0           0.0         0.0        0.0  ...            5.837          -69.0          -58.0          94125.0                    14.65         Sal
n_g5

[5 rows x 189 columns]
tree :
0.9512168856342882
bagging :
0.9837389618780961
```

Random Forest

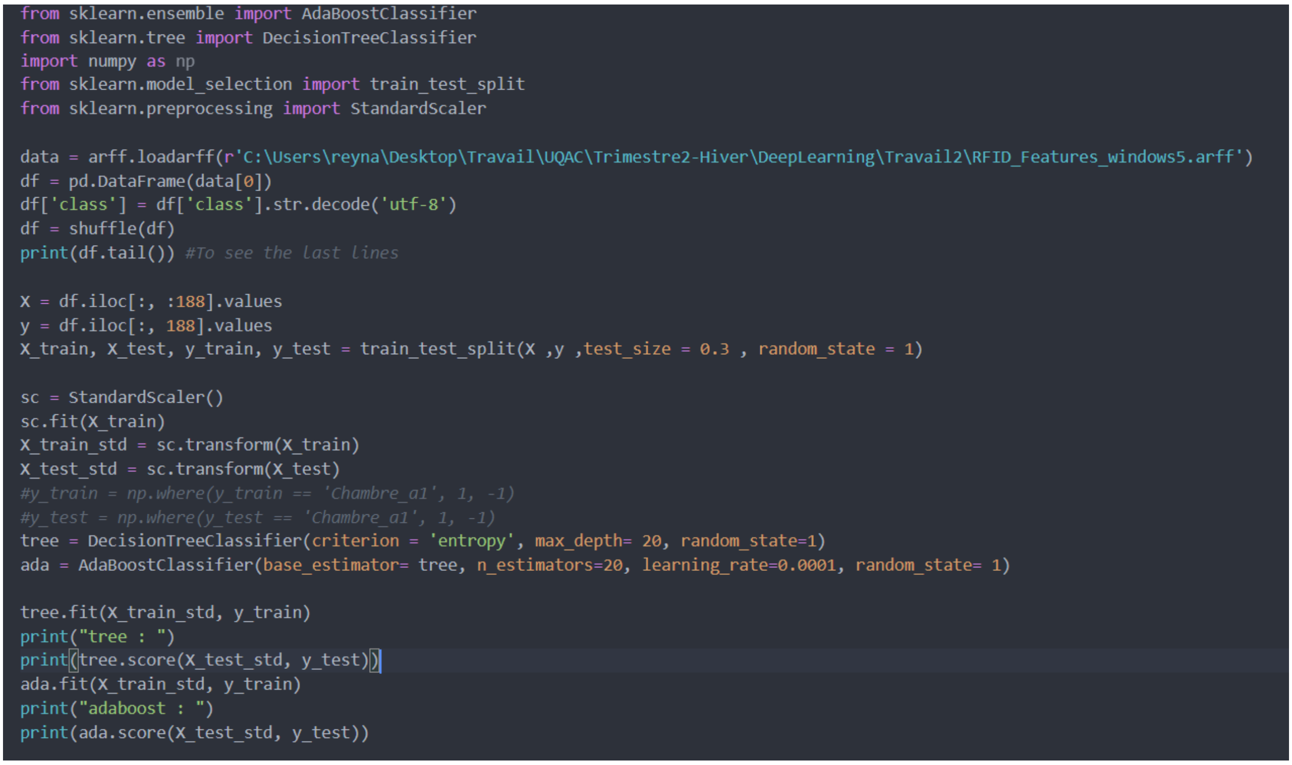


```
from sklearn.ensemble import RandomForestClassifier
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

data = arff.loadarff(r'C:\Users\reyna\Desktop\Travail\UQAC\Trimestre2-Hiver\DeepLearning\Travail2\RFID_Features_windows5.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df = shuffle(df)
print(df.tail()) #To see the last lines

X = df.iloc[:, :188].values
y = df.iloc[:, 188].values
X_train, X_test, y_train, y_test = train_test_split(X ,y ,test_size = 0.3 , random_state = 1)

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
#y_train = np.where(y_train == 'Chambre_a1', 1, -1)
#y_test = np.where(y_test == 'Chambre_a1', 1, -1)

forest = RandomForestClassifier(criterion = 'entropy', n_estimators=10, n_jobs = 2, random_state=1)

forest.fit(X_train_std, y_train)
print("random forest : ")
print(forest.score(X_test_std, y_test))
```

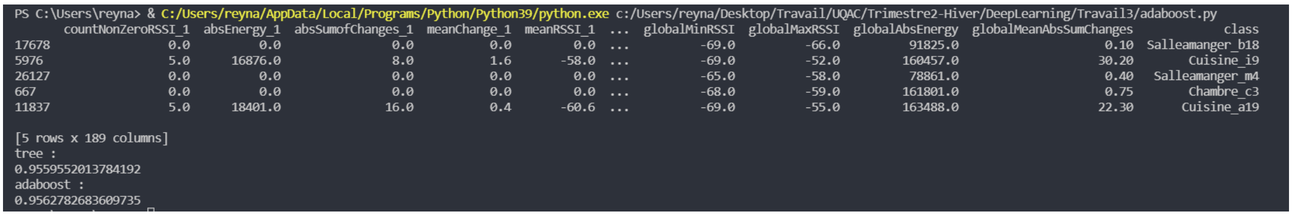


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail3/randomforest.py
       countNonZeroRSSI_1  absEnergy_1  absSumofChanges_1  meanChange_1  meanRSSI_1  minRSSI_1  ...  globalMeanStDev  globalMinRSSI  globalMaxRSSI  globalAbsEnergy  globalMeanAbsSumChanges            c
lass
20474                 0.0          0.0                0.0           0.0         0.0        0.0  ...            0.083          -66.0          -59.0          79113.0                     0.35  Salleamange
r_f7
343                   0.0          0.0                0.0           0.0         0.0        0.0  ...            0.222          -67.0          -56.0         150466.0                     0.80       Chambr
e_b5
15739                 0.0          0.0                0.0           0.0         0.0        0.0  ...            0.136          -69.0          -58.0         165322.0                     0.45         Salo
n_f2
1384                  0.0          0.0                0.0           0.0         0.0        0.0  ...            0.188          -68.0          -53.0         156625.0                     0.55       Chambr
e_e0
2242                  5.0      17298.0                9.0          -0.6       -58.8      -61.0  ...            0.187          -66.0          -45.0          66012.0                     0.75       Cuisin
e_e2

[5 rows x 189 columns]
random forest :
0.9854619857850527
```

Adaboost

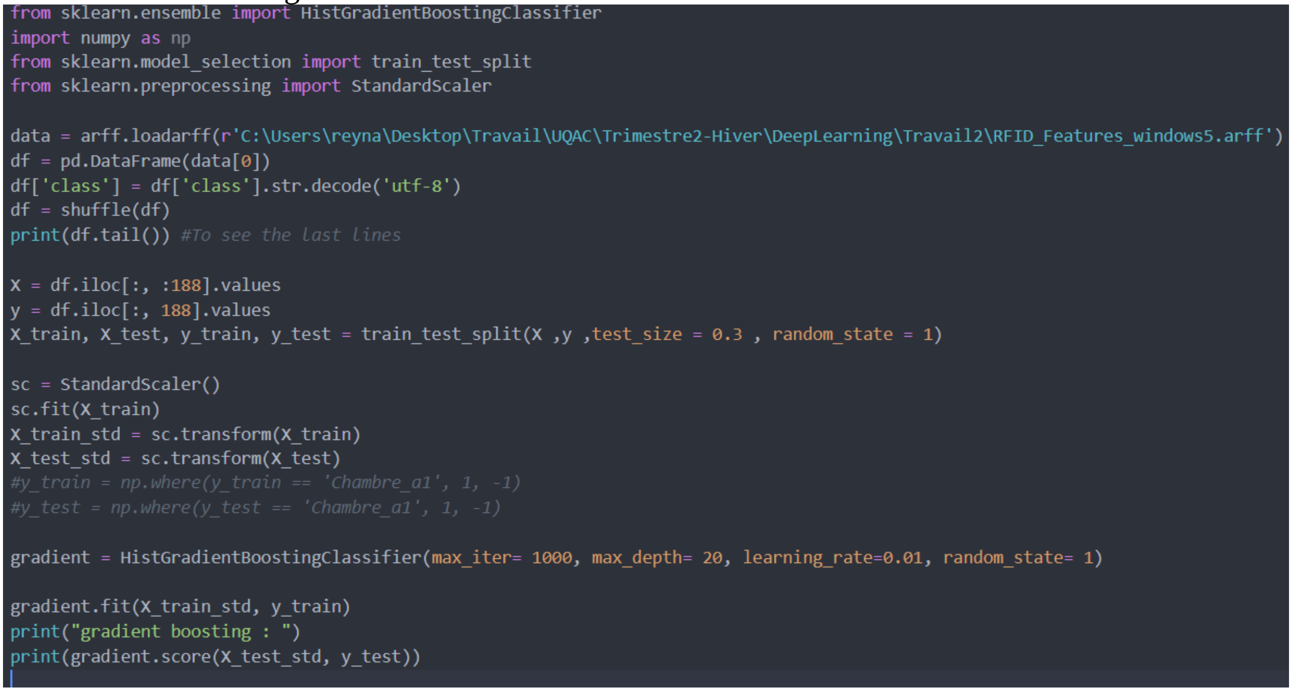


```python
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

data = arff.loadarff(r'C:\Users\reyna\Desktop\Travail\UQAC\Trimestre2-Hiver\DeepLearning\Travail2\RFID_Features_windows5.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df = shuffle(df)
print(df.tail()) #To see the last lines

X = df.iloc[:, :188].values
y = df.iloc[:, 188].values
X_train, X_test, y_train, y_test = train_test_split(X ,y ,test_size = 0.3 , random_state = 1)

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
#y_train = np.where(y_train == 'Chambre_a1', 1, -1)
#y_test = np.where(y_test == 'Chambre_a1', 1, -1)
tree = DecisionTreeClassifier(criterion = 'entropy', max_depth= 20, random_state=1)
ada = AdaBoostClassifier(base_estimator= tree, n_estimators=20, learning_rate=0.0001, random_state= 1)

tree.fit(X_train_std, y_train)
print("tree : ")
print(tree.score(X_test_std, y_test))
ada.fit(X_train_std, y_train)
print("adaboost : ")
print(ada.score(X_test_std, y_test))
```

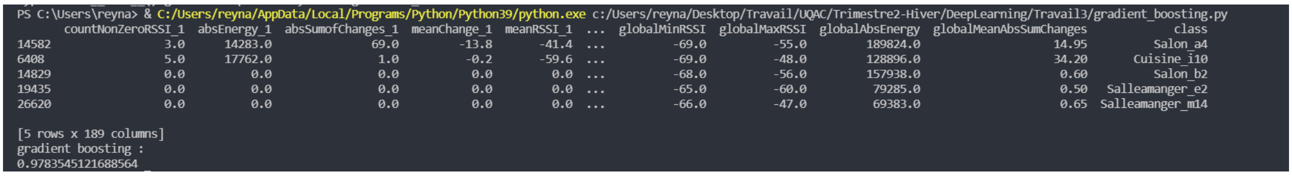


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail3/adaboost.py
       countNonZeroRSSI_1  absEnergy_1  absSumofChanges_1  meanChange_1  meanRSSI_1  ...  globalMinRSSI  globalMaxRSSI  globalAbsEnergy  globalMeanAbsSumChanges             class
17678                 0.0          0.0                0.0           0.0         0.0  ...          -69.0          -66.0          91825.0                     0.10  Salleamanger_b18
5976                  5.0      16876.0                8.0           1.6       -58.0  ...          -69.0          -52.0         160457.0                    30.20        Cuisine_i9
26127                 0.0          0.0                0.0           0.0         0.0  ...          -65.0          -58.0          78861.0                     0.40   Salleamanger_m4
667                   0.0          0.0                0.0           0.0         0.0  ...          -68.0          -59.0         161801.0                     0.75        Chambre_c3
11837                 5.0      18401.0               16.0           0.4       -60.6  ...          -69.0          -55.0         163488.0                    22.30       Cuisine_a19

[5 rows x 189 columns]
tree :
0.9559552013784192
adaboost :
0.9562782683609735
```

Hist Gradient Boosting

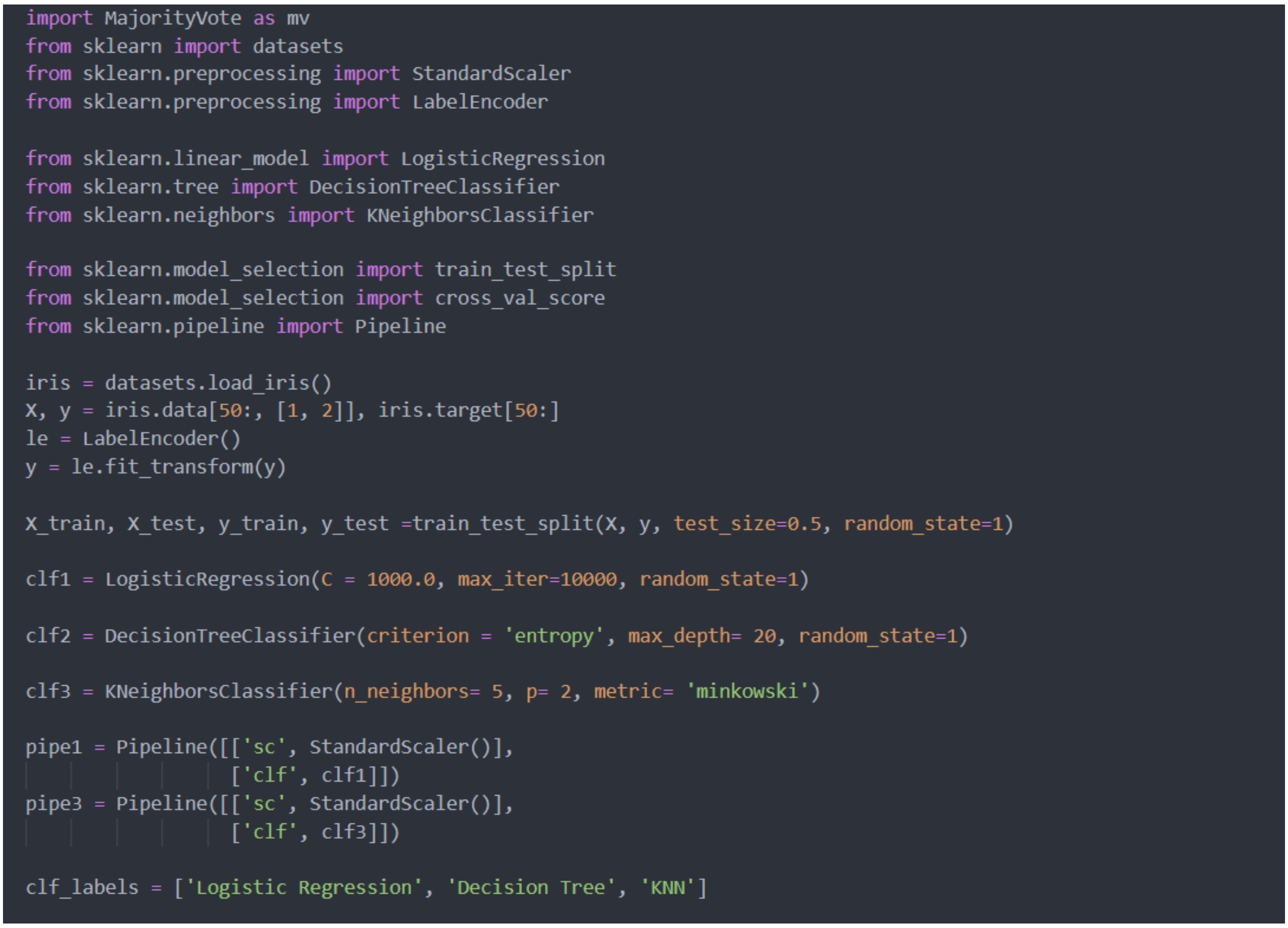


```
from sklearn.ensemble import HistGradientBoostingClassifier
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

data = arff.loadarff(r'C:\Users\reyna\Desktop\Travail\UQAC\Trimestre2-Hiver\DeepLearning\Travail2\RFID_Features_windows5.arff')
df = pd.DataFrame(data[0])
df['class'] = df['class'].str.decode('utf-8')
df = shuffle(df)
print(df.tail()) #To see the last lines

X = df.iloc[:, :188].values
y = df.iloc[:, 188].values
X_train, X_test, y_train, y_test = train_test_split(X ,y ,test_size = 0.3 , random_state = 1)

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
#y_train = np.where(y_train == 'Chambre_a1', 1, -1)
#y_test = np.where(y_test == 'Chambre_a1', 1, -1)

gradient = HistGradientBoostingClassifier(max_iter= 1000, max_depth= 20, learning_rate=0.01, random_state= 1)

gradient.fit(X_train_std, y_train)
print("gradient boosting : ")
print(gradient.score(X_test_std, y_test))
```

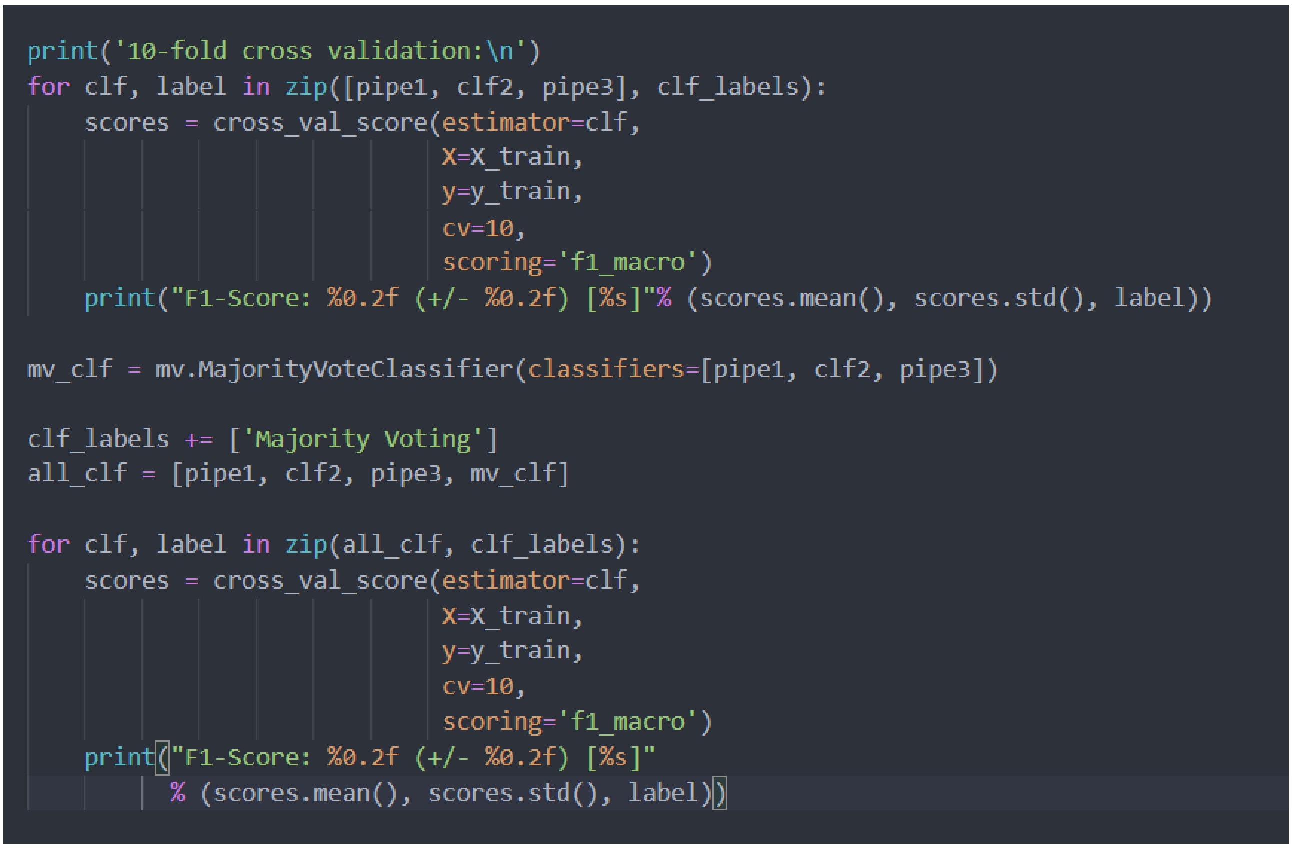


```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail3/gradient_boosting.py
       countNonZeroRSSI_1  absEnergy_1  absSumofChanges_1  meanChange_1  meanRSSI_1  ...  globalMinRSSI  globalMaxRSSI  globalAbsEnergy  globalMeanAbsSumChanges            class
14582                 3.0      14283.0               69.0         -13.8       -41.4  ...          -69.0          -55.0         189824.0                    14.95         Salon_a4
6408                  5.0      17762.0                1.0          -0.2       -59.6  ...          -69.0          -48.0         128896.0                    34.20      Cuisine_i10
14829                 0.0          0.0                0.0           0.0         0.0  ...          -68.0          -56.0         157938.0                     0.60         Salon_b2
19435                 0.0          0.0                0.0           0.0         0.0  ...          -65.0          -60.0          79285.0                     0.50   Salleamanger_e2
26620                 0.0          0.0                0.0           0.0         0.0  ...          -66.0          -47.0          69383.0                     0.65  Salleamanger_m14

[5 rows x 189 columns]
gradient boosting :
0.9783545121688564
```

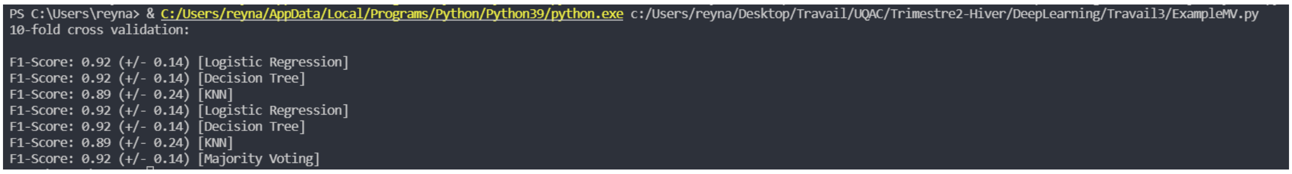
MajorityVote



```python
import MajorityVote as mv
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.pipeline import Pipeline

iris = datasets.load_iris()
X, y = iris.data[50:, [1, 2]], iris.target[50:]
le = LabelEncoder()
y = le.fit_transform(y)

X_train, X_test, y_train, y_test =train_test_split(X, y, test_size=0.5, random_state=1)

clf1 = LogisticRegression(C = 1000.0, max_iter=10000, random_state=1)

clf2 = DecisionTreeClassifier(criterion = 'entropy', max_depth= 20, random_state=1)

clf3 = KNeighborsClassifier(n_neighbors= 5, p= 2, metric= 'minkowski')

pipe1 = Pipeline([['sc', StandardScaler()],
                  ['clf', clf1]])
pipe3 = Pipeline([['sc', StandardScaler()],
                  ['clf', clf3]])

clf_labels = ['Logistic Regression', 'Decision Tree', 'KNN']
```



```python
print('10-fold cross validation:\n')
for clf, label in zip([pipe1, clf2, pipe3], clf_labels):
    scores = cross_val_score(estimator=clf,
                             X=X_train,
                             y=y_train,
                             cv=10,
                             scoring='f1_macro')
    print("F1-Score: %0.2f (+/- %0.2f) [%s]"% (scores.mean(), scores.std(), label))

mv_clf = mv.MajorityVoteClassifier(classifiers=[pipe1, clf2, pipe3])

clf_labels += ['Majority Voting']
all_clf = [pipe1, clf2, pipe3, mv_clf]

for clf, label in zip(all_clf, clf_labels):
    scores = cross_val_score(estimator=clf,
                             X=X_train,
                             y=y_train,
                             cv=10,
                             scoring='f1_macro')
    print("F1-Score: %0.2f (+/- %0.2f) [%s]"
          % (scores.mean(), scores.std(), label))
```



```
PS C:\Users\reyna> & C:/Users/reyna/AppData/Local/Programs/Python/Python39/python.exe c:/Users/reyna/Desktop/Travail/UQAC/Trimestre2-Hiver/DeepLearning/Travail3/ExampleMV.py
10-fold cross validation:

F1-Score: 0.92 (+/- 0.14) [Logistic Regression]
F1-Score: 0.92 (+/- 0.14) [Decision Tree]
F1-Score: 0.89 (+/- 0.24) [KNN]
F1-Score: 0.92 (+/- 0.14) [Logistic Regression]
F1-Score: 0.92 (+/- 0.14) [Decision Tree]
F1-Score: 0.89 (+/- 0.24) [KNN]
F1-Score: 0.92 (+/- 0.14) [Majority Voting]
```