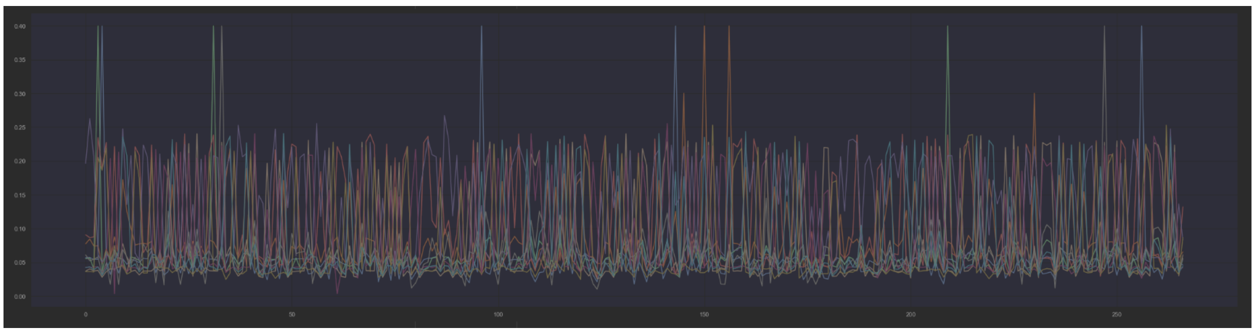
Dragchicescu Adrian-Ioan 343CB

Report Tema Invatare Automata

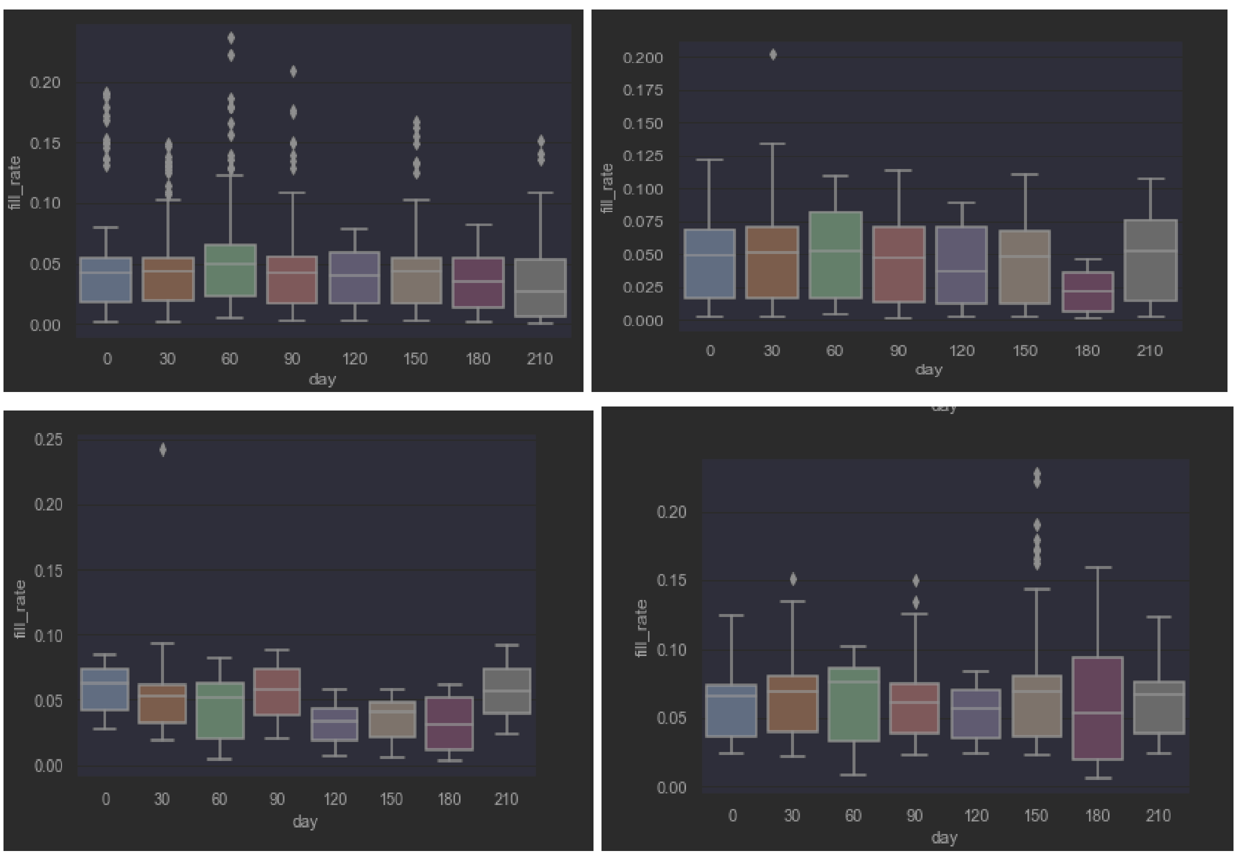
Explorarea si vizualizarea datelor

PEMS

Media zilnica a top 10 sensori dupa deviatia standard a medie pe parcursul unui an

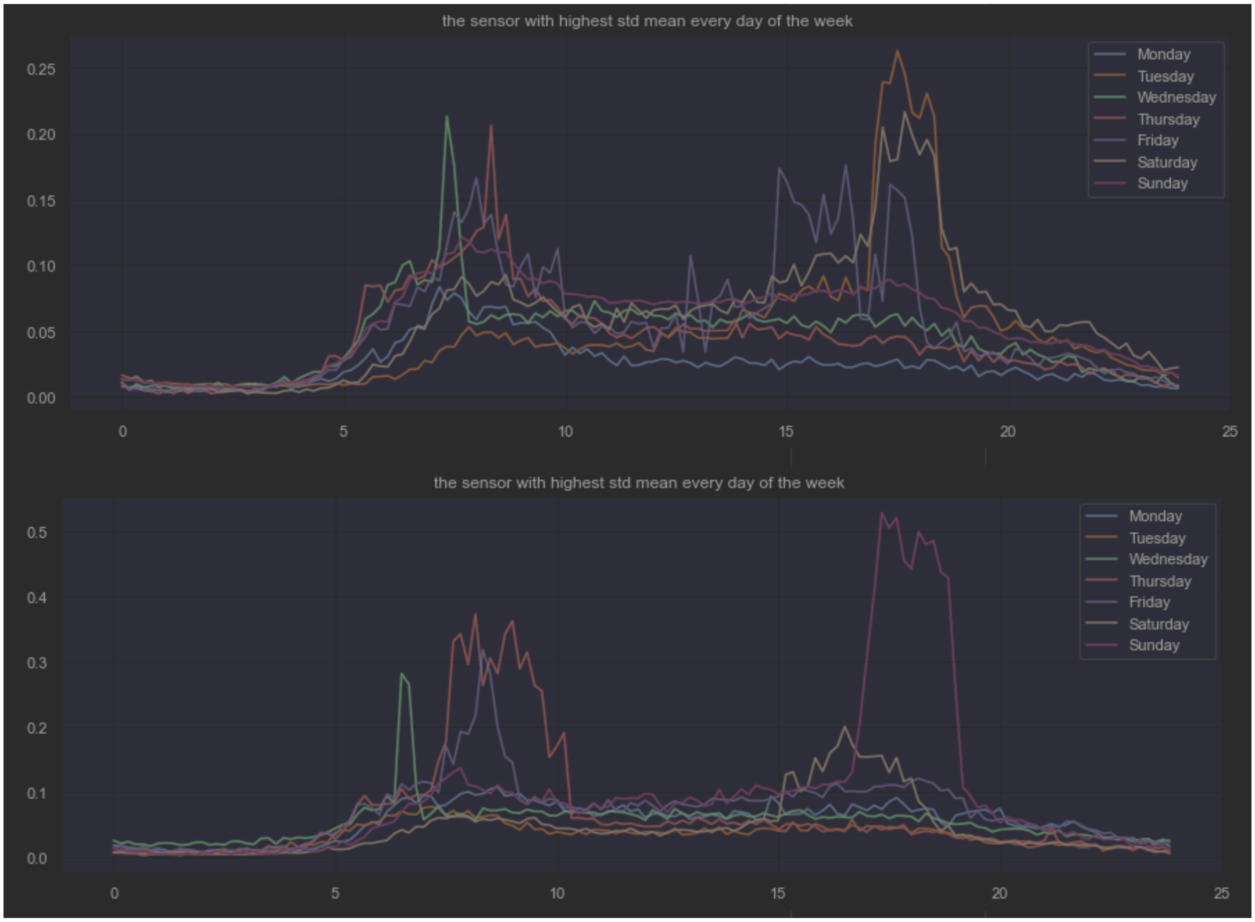


Rata de umplere pentru primii 4 senzori



Se poate observa ca media de umlere a starzilor este destul de mica in jur de 0.05 iar zilele nu difera foarte mult (fiind plotate datele unui zile din 30 in 30 de zile se poate arata ca in cele 8 exemple avem cel putin un exemplu din fiecare zi)

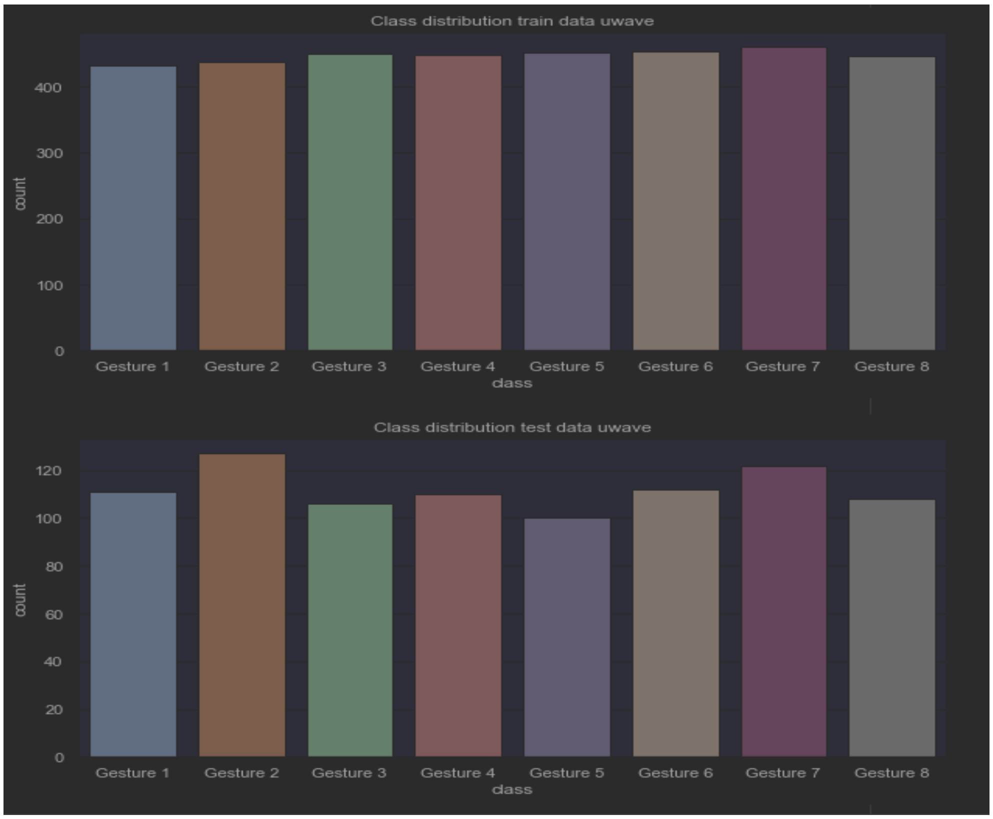
Senzori pe zile:



Se poate vedea ca cele mai importante intervale (cele mai active si cele mai cu diferente cele mai are de la o zi la alta) sunt de la 6 la 10 si de la 15 la 20 aproximativ deci niste atribute uitle ar fi caracteristicele din acele intervale

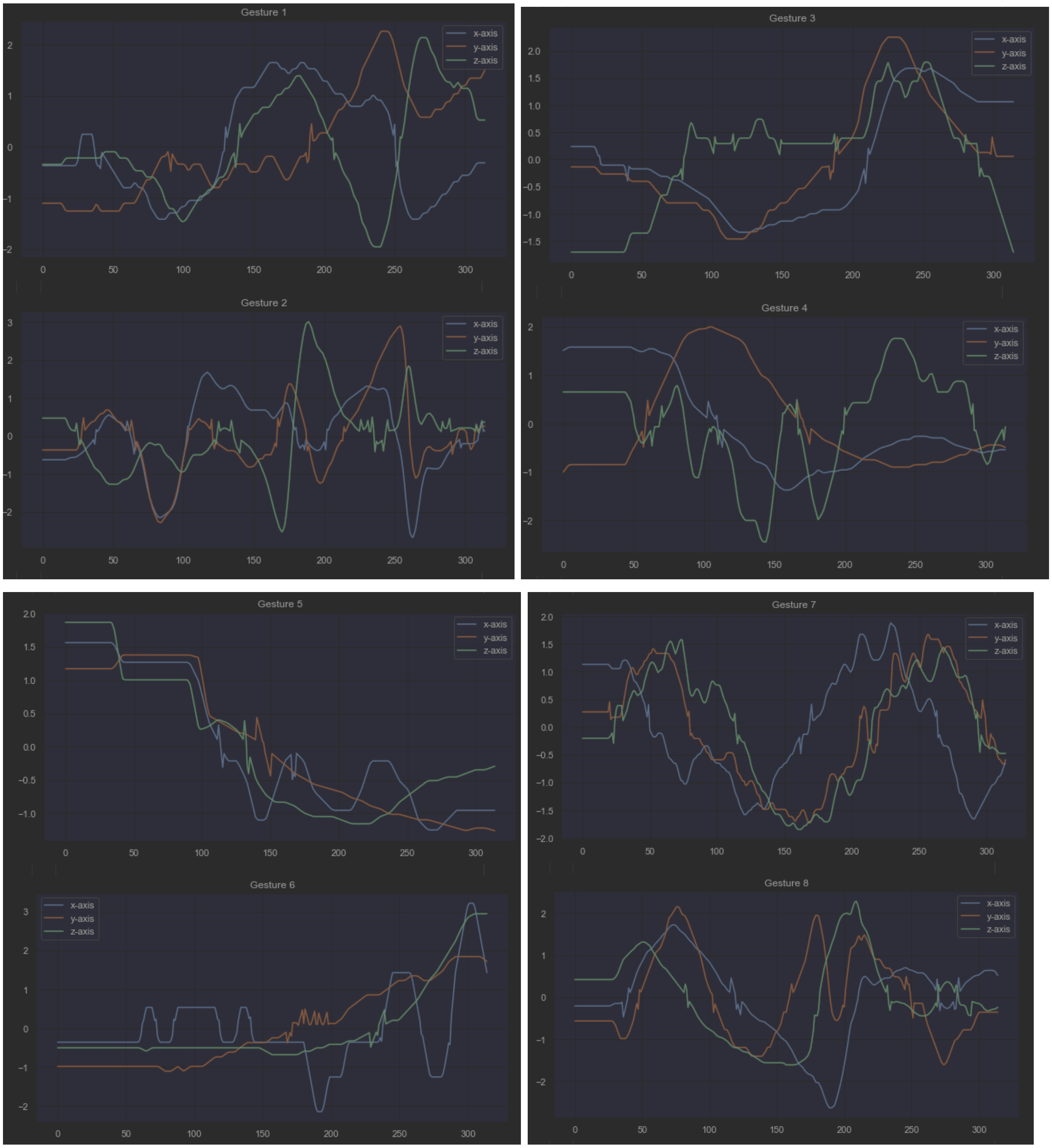
UWave:

Distributia claselor



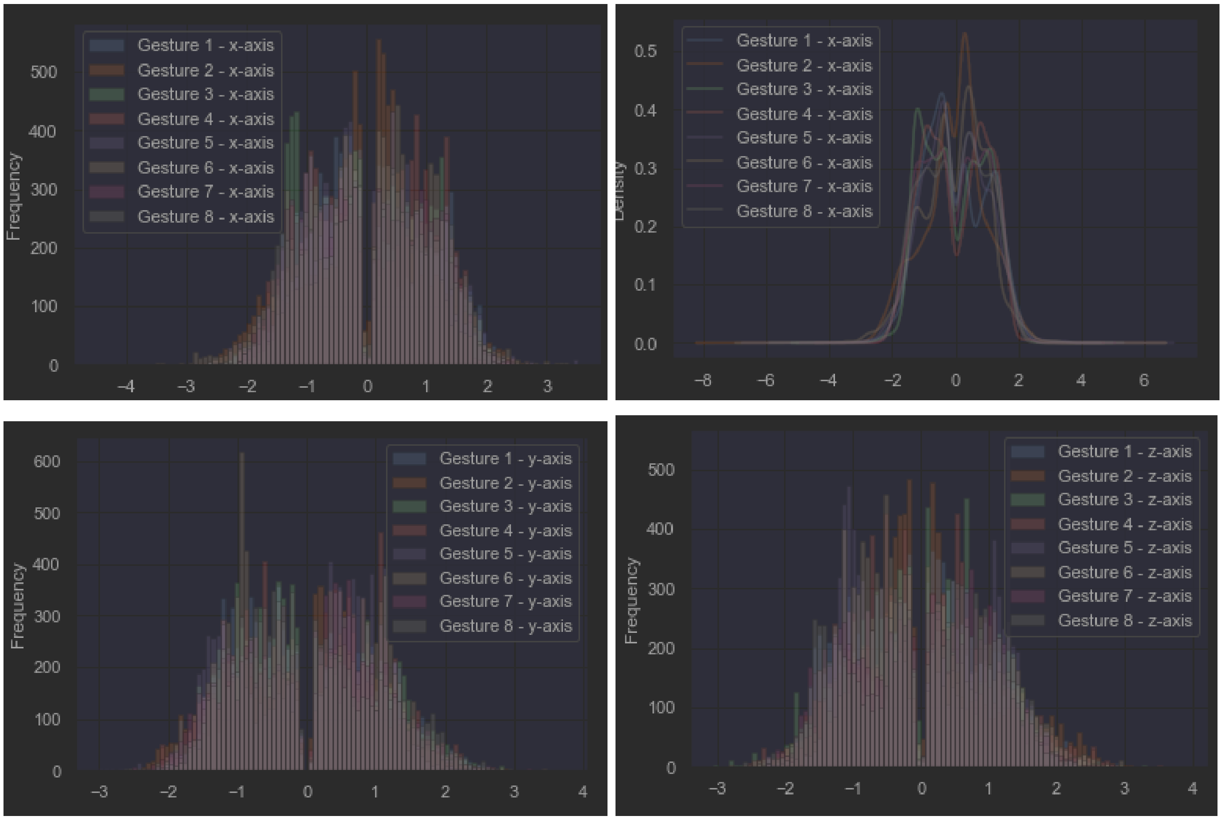
Se poate observa ca avem clase echilibrate atat pentru train cat si pentru test.

Exemple de serii temporale pe cele 3 axe pentru fiecare gest:



Se poate observa ca majoritatea seriilor acopera toata distribuita de la -2 la 2 (min si max pe intraga serise nu ar fi foarte util de util)

Distribuitia



Algorimi si extragerea atributelor:

Am procesat datele in trei moduluri (si am antreant modelel pe fiecare tip):

1. Le am normalizat si le am lasat cate atribute aveau initial
2. Am folosit VarianceThreshold si SelectPercentile sa reduc numarul de atribute automat
3. Am extras manual un numar de atribute pentru fiecare si am aplicat ulterior VarianceThreshold.

UWave:

1. Am folosit seltul de date pentru test (cel de 3k exemple) pentru antrenare si cel dat pt train pentru test.Setul folosit pentru antrenare are 945 atribute.
2. Am folosit VarianceThreshold cu threshold=0.995 si SelectPercentile cu percentile=20 rezultand in 105 atribute.
3. Am calculate pentru fiecare fereastra de 105 a fiecarei axa urmatoarele caracteristici:
    a. Medie
    b. Abaterea standard
    c. Abaterea medie absolută
    d. Valoare minimă
    e. Valoare maximă
    f. Diferenta de valori maxime si minime

g. Mediană
h. abaterea mediană absolută
i. intervalul intercuartil
j. Număr de valori negative
k. Număr de valori pozitive
l. Număr de valori peste medie
m. Număr de vârfuri
n. Energia semnalului
o. Asimetrie (skewness)
p. Curtoză (kurtosis)
q. Accelerația medie rezultantă

Si pentru datele propriu zise cat si pentru datele in regim de frecventa

Resultand 144 de atribute dupa care am folsit VarianceThreshold cu threshold=0.5 si SelectPercentile cu percentile=50 rezultand in 96 atribute.

RandomForest

1.Best parameters set found on development set:
{'max_depth': 100, 'n_estimators': 1000} cu accuracy 0.96

Am testat uramtoarele combinatii:
0.914 (+/-0.010) for {'max_depth': 10, 'max_samples': 0.1, 'n_estimators': 100}
0.926 (+/-0.006) for {'max_depth': 10, 'max_samples': 0.1, 'n_estimators': 1000}
0.933 (+/-0.012) for {'max_depth': 10, 'max_samples': 0.2, 'n_estimators': 100}
0.938 (+/-0.015) for {'max_depth': 10, 'max_samples': 0.2, 'n_estimators': 1000}
0.948 (+/-0.015) for {'max_depth': 10, 'max_samples': 0.5, 'n_estimators': 100}
0.951 (+/-0.016) for {'max_depth': 10, 'max_samples': 0.5, 'n_estimators': 1000}
0.920 (+/-0.008) for {'max_depth': 100, 'max_samples': 0.1, 'n_estimators': 100}
0.926 (+/-0.011) for {'max_depth': 100, 'max_samples': 0.1, 'n_estimators': 1000}
0.937 (+/-0.013) for {'max_depth': 100, 'max_samples': 0.2, 'n_estimators': 100}
0.940 (+/-0.014) for {'max_depth': 100, 'max_samples': 0.2, 'n_estimators': 1000}
0.949 (+/-0.012) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 100}
0.951 (+/-0.018) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 1000}
0.925 (+/-0.006) for {'max_depth': 500, 'max_samples': 0.1, 'n_estimators': 100}
0.926 (+/-0.014) for {'max_depth': 500, 'max_samples': 0.1, 'n_estimators': 1000}
0.933 (+/-0.011) for {'max_depth': 500, 'max_samples': 0.2, 'n_estimators': 100}
0.941 (+/-0.016) for {'max_depth': 500, 'max_samples': 0.2, 'n_estimators': 1000}
0.953 (+/-0.016) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 100}
0.950 (+/-0.016) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 1000}
0.949 (+/-0.017) for {'max_depth': 10, 'n_estimators': 100}
0.954 (+/-0.016) for {'max_depth': 10, 'n_estimators': 1000}
0.954 (+/-0.008) for {'max_depth': 100, 'n_estimators': 100}
0.956 (+/-0.013) for {'max_depth': 100, 'n_estimators': 1000}
0.953 (+/-0.016) for {'max_depth': 500, 'n_estimators': 100}

0.956 (+/-0.014) for {'max_depth': 500, 'n_estimators': 1000}

2. Best parameters set found on development set:
{'max_depth': 100, 'n_estimators': 1000} cu accuracy0.832

0.813 (+/-0.015) for {'max_depth': 10, 'max_samples': 0.5, 'n_estimators': 100}
0.813 (+/-0.021) for {'max_depth': 10, 'max_samples': 0.5, 'n_estimators': 1000}
0.815 (+/-0.012) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 100}
0.821 (+/-0.011) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 1000}
0.819 (+/-0.007) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 100}
0.817 (+/-0.012) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 1000}
0.814 (+/-0.020) for {'max_depth': 10, 'n_estimators': 100}
0.819 (+/-0.017) for {'max_depth': 10, 'n_estimators': 1000}
0.826 (+/-0.009) for {'max_depth': 100, 'n_estimators': 100}
0.831 (+/-0.006) for {'max_depth': 100, 'n_estimators': 1000}
0.826 (+/-0.017) for {'max_depth': 500, 'n_estimators': 100}
0.829 (+/-0.007) for {'max_depth': 500, 'n_estimators': 1000}

3. Best parameters set found on development set:
{'max_depth': 500, 'n_estimators': 1000} cu accuracy **0.967**

0.948 (+/-0.008) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 100}
0.952 (+/-0.006) for {'max_depth': 100, 'max_samples': 0.5, 'n_estimators': 1000}
0.950 (+/-0.005) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 100}
0.951 (+/-0.002) for {'max_depth': 500, 'max_samples': 0.5, 'n_estimators': 1000}
0.958 (+/-0.012) for {'max_depth': 100, 'n_estimators': 100}
0.956 (+/-0.007) for {'max_depth': 100, 'n_estimators': 200}
0.957 (+/-0.009) for {'max_depth': 100, 'n_estimators': 1000}
0.954 (+/-0.005) for {'max_depth': 200, 'n_estimators': 100}
0.958 (+/-0.007) for {'max_depth': 200, 'n_estimators': 200}
0.958 (+/-0.008) for {'max_depth': 200, 'n_estimators': 1000}
0.957 (+/-0.007) for {'max_depth': 500, 'n_estimators': 100}
0.957 (+/-0.008) for {'max_depth': 500, 'n_estimators': 200}
0.960 (+/-0.008) for {'max_depth': 500, 'n_estimators': 1000}

Matricea de confuzie:

```
[[117   0   0   1   1   3   0   0]
 [  0 107   0   0   0   0   1   0]
 [  0   0 105   0   1   0   0   0]
 [  1   0   0 106   2   0   0   1]
 [  0   0   1   1 123   1   0   1]
 [  1   0   2   7   1 100   0   0]
 [  0   0   1   0   0   0 111   0]
```

[1   0   1   0   0   0   0 98]]

SVC:

1.Best parameters set found on development set:
{'C': 5, 'gamma': 0.001, 'kernel': 'rbf'} cu accuracy **0.982**

0.950 (+/-0.007) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.892 (+/-0.010) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'linear'}
0.945 (+/-0.008) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'poly'}
0.891 (+/-0.009) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.892 (+/-0.010) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'poly'}
0.964 (+/-0.003) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.881 (+/-0.010) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'linear'}
0.958 (+/-0.008) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'poly'}
0.910 (+/-0.006) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.881 (+/-0.010) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'poly'}
0.970 (+/-0.005) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.874 (+/-0.018) for {'C': 1, 'gamma': 0.001, 'kernel': 'linear'}
0.962 (+/-0.008) for {'C': 1, 'gamma': 0.001, 'kernel': 'poly'}
0.926 (+/-0.006) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.874 (+/-0.018) for {'C': 1, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 1, 'gamma': 0.0001, 'kernel': 'poly'}
0.975 (+/-0.006) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.867 (+/-0.018) for {'C': 5, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.007) for {'C': 5, 'gamma': 0.001, 'kernel': 'poly'}
0.949 (+/-0.008) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.867 (+/-0.018) for {'C': 5, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 5, 'gamma': 0.0001, 'kernel': 'poly'}
0.975 (+/-0.010) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.867 (+/-0.018) for {'C': 10, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.012) for {'C': 10, 'gamma': 0.001, 'kernel': 'poly'}
0.953 (+/-0.013) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.867 (+/-0.018) for {'C': 10, 'gamma': 0.0001, 'kernel': 'linear'}
0.867 (+/-0.018) for {'C': 10, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.012) for {'C': 10, 'gamma': 0.001, 'kernel': 'poly'}
0.953 (+/-0.013) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.867 (+/-0.018) for {'C': 10, 'gamma': 0.0001, 'kernel': 'linear'}
0.612 (+/-0.037) for {'C': 10, 'gamma': 0.0001, 'kernel': 'poly'}

Matricea de confuzie:

```
[[119   0   0   1   0   2   0   0]
 [  0 108   0   0   0   0   0   0]
 [  0   0 106   0   0   0   0   0]
 [  0   0   0 105   4   0   0   1]
 [  0   0   1   1 125   0   0   0]
 [  0   0   1   4   1 105   0   0]
 [  0   0   0   0   0   0 112   0]
 [  0   0   0   0   0   0   0 100]]
```

2. Best parameters set found on development set:
{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'} cu accuracy 0.756

0.682 (+/-0.023) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.727 (+/-0.007) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'poly'}
0.569 (+/-0.028) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.727 (+/-0.007) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'poly'}
0.696 (+/-0.015) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.730 (+/-0.013) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'linear'}
0.133 (+/-0.003) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'poly'}
0.643 (+/-0.024) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.730 (+/-0.013) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'poly'}
0.707 (+/-0.019) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.726 (+/-0.009) for {'C': 1, 'gamma': 0.001, 'kernel': 'linear'}
0.173 (+/-0.009) for {'C': 1, 'gamma': 0.001, 'kernel': 'poly'}
0.670 (+/-0.019) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.726 (+/-0.009) for {'C': 1, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 1, 'gamma': 0.0001, 'kernel': 'poly'}
0.737 (+/-0.014) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.714 (+/-0.020) for {'C': 5, 'gamma': 0.001, 'kernel': 'linear'}
0.513 (+/-0.018) for {'C': 5, 'gamma': 0.001, 'kernel': 'poly'}
0.692 (+/-0.012) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.714 (+/-0.020) for {'C': 5, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 5, 'gamma': 0.0001, 'kernel': 'poly'}
0.759 (+/-0.017) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.708 (+/-0.012) for {'C': 10, 'gamma': 0.001, 'kernel': 'linear'}
0.575 (+/-0.033) for {'C': 10, 'gamma': 0.001, 'kernel': 'poly'}
0.699 (+/-0.014) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.708 (+/-0.012) for {'C': 10, 'gamma': 0.0001, 'kernel': 'linear'}
0.128 (+/-0.000) for {'C': 10, 'gamma': 0.0001, 'kernel': 'poly'}

3. Best parameters set found on development set:
{'C': 0.2, 'gamma': 0.001, 'kernel': 'poly'} cu accuracy 0.965

0.128 (+/-0.000) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'rbf'}
0.935 (+/-0.009) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 0.2, 'gamma': 0.001, 'kernel': 'poly'}
0.928 (+/-0.016) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'rbf'}
0.935 (+/-0.009) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 0.2, 'gamma': 0.0001, 'kernel': 'poly'}
0.129 (+/-0.001) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'rbf'}
0.931 (+/-0.005) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 0.5, 'gamma': 0.001, 'kernel': 'poly'}
0.948 (+/-0.003) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.931 (+/-0.005) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 0.5, 'gamma': 0.0001, 'kernel': 'poly'}
0.356 (+/-0.010) for {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
0.929 (+/-0.006) for {'C': 1, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 1, 'gamma': 0.001, 'kernel': 'poly'}
0.962 (+/-0.004) for {'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'}
0.929 (+/-0.006) for {'C': 1, 'gamma': 0.0001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 1, 'gamma': 0.0001, 'kernel': 'poly'}
0.390 (+/-0.008) for {'C': 5, 'gamma': 0.001, 'kernel': 'rbf'}
0.928 (+/-0.005) for {'C': 5, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 5, 'gamma': 0.001, 'kernel': 'poly'}
0.963 (+/-0.005) for {'C': 5, 'gamma': 0.0001, 'kernel': 'rbf'}
0.928 (+/-0.005) for {'C': 5, 'gamma': 0.0001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 5, 'gamma': 0.0001, 'kernel': 'poly'}
0.390 (+/-0.008) for {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
0.928 (+/-0.005) for {'C': 10, 'gamma': 0.001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 10, 'gamma': 0.001, 'kernel': 'poly'}
0.963 (+/-0.005) for {'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}
0.928 (+/-0.005) for {'C': 10, 'gamma': 0.0001, 'kernel': 'linear'}
0.967 (+/-0.005) for {'C': 10, 'gamma': 0.0001, 'kernel': 'poly'}

GradientBoosted Trees

1. Best parameters set found on development set:
{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200'} cu accuracy 0.959

0.957 (+/-0.016) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
0.957 (+/-0.015) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200'}
0.952 (+/-0.016) for {'learning_rate': 0.2, 'max_depth': 10, 'n_estimators': 100}

0.954 (+/-0.015) for {'learning_rate': 0.2, 'max_depth': 10, 'n_estimators': 200}
0.955 (+/-0.020) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
0.956 (+/-0.018) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
0.951 (+/-0.016) for {'learning_rate': 0.3, 'max_depth': 10, 'n_estimators': 100}
0.949 (+/-0.017) for {'learning_rate': 0.3, 'max_depth': 10, 'n_estimators': 200}
0.954 (+/-0.013) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}
0.954 (+/-0.013) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 200}
0.952 (+/-0.013) for {'learning_rate': 0.5, 'max_depth': 10, 'n_estimators': 100}
0.953 (+/-0.009) for {'learning_rate': 0.5, 'max_depth': 10, 'n_estimators': 200}

2. Best parameters set found on development set:
{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300} cu accuracy **0.84**

0.696 (+/-0.021) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
0.698 (+/-0.024) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
0.697 (+/-0.023) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
0.696 (+/-0.021) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 100}
0.698 (+/-0.024) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 200}
0.697 (+/-0.023) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 300}
0.700 (+/-0.024) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
0.697 (+/-0.025) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
0.698 (+/-0.024) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 300}
0.700 (+/-0.024) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 100}
0.697 (+/-0.025) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 200}
0.698 (+/-0.024) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 300}
0.697 (+/-0.028) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}
0.695 (+/-0.031) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 200}
0.694 (+/-0.031) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 300}
0.697 (+/-0.028) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 100}
0.695 (+/-0.031) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 200}
0.694 (+/-0.031) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 300}
0.845 (+/-0.010) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
0.847 (+/-0.015) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
0.850 (+/-0.019) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
0.844 (+/-0.024) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 100}
0.845 (+/-0.027) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 200}
0.844 (+/-0.030) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 300}
0.843 (+/-0.012) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
0.844 (+/-0.014) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
0.842 (+/-0.014) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 300}
0.841 (+/-0.018) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 100}
0.843 (+/-0.017) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 200}
0.842 (+/-0.019) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 300}
0.839 (+/-0.018) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}

0.840 (+/-0.019) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 200}
0.840 (+/-0.018) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 300}
0.836 (+/-0.022) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 100}
0.836 (+/-0.021) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 200}
0.836 (+/-0.019) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 300}

Matricea de confuzie:

```
[[ 87   0   9   2   0  22   0   2]
 [  0 105   0   0   0   0   2   1]
 [  7   0  81   0   5   1  12   0]
 [  1   0   1  95   9   2   0   2]
 [  0   0   4   8 114   0   0   1]
 [ 20   0   1   6   0  83   0   1]
 [  3   2   9   1   0   1  96   0]
 [  0   1   2   2   1   0   2  92]]
```

3. Best parameters set found on development set:
{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100} cu accuracy de 0.967

0.946 (+/-0.011) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
0.948 (+/-0.009) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
0.947 (+/-0.010) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
0.946 (+/-0.011) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 100}
0.948 (+/-0.009) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 200}
0.947 (+/-0.010) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 300}
0.948 (+/-0.008) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
0.948 (+/-0.012) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
0.945 (+/-0.007) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 300}
0.948 (+/-0.008) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 100}
0.948 (+/-0.012) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 200}
0.945 (+/-0.007) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 300}
0.949 (+/-0.010) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}
0.944 (+/-0.009) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 200}
0.943 (+/-0.006) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 300}
0.949 (+/-0.010) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 100}
0.944 (+/-0.009) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 200}
0.943 (+/-0.006) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 300}
0.960 (+/-0.004) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
0.960 (+/-0.008) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
0.961 (+/-0.007) for {'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
0.960 (+/-0.011) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 100}
0.961 (+/-0.011) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 200}
0.961 (+/-0.010) for {'learning_rate': 0.2, 'max_depth': 15, 'n_estimators': 300}

0.962 (+/-0.008) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
0.961 (+/-0.009) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
0.961 (+/-0.007) for {'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 300}
0.960 (+/-0.005) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 100}
0.960 (+/-0.007) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 200}
0.960 (+/-0.007) for {'learning_rate': 0.3, 'max_depth': 15, 'n_estimators': 300}
0.960 (+/-0.007) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 100}
0.960 (+/-0.006) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 200}
0.960 (+/-0.006) for {'learning_rate': 0.5, 'max_depth': 5, 'n_estimators': 300}
0.960 (+/-0.008) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 100}
0.959 (+/-0.006) for {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 200}