	Precision	Recall	F1-Score	Support
SVM DEFAULT PARAMETERS, ALL FEATURES				
badminton_clear	0.4	0.67	0.5	43
badminton_smash	0.58	0.38	0.45	40
squash_backhandboast	0.61	0.65	0.63	34
squash_forehandboast	0.59	0.29	0.38	35
Accuracy	0.54	0.5	0.49	152
SVM (C=0.1, gamma=1, kernel='poly') GridSearchCV , ALL FEATURES				
badminton_clear	0.62	0.65	0.64	43
badminton_smash	0.62	0.6	0.61	40
squash_backhandboast	0.88	0.68	0.77	34
squash_forehandboast	0.69	0.83	0.75	35
Accuracy	0.7	0.69	0.69	152
SVM (C=0.1, gamma=1, kernel='poly') GridSearchCV, VARIANCE TRESHOLD=0.25				
badminton_clear	0.62	0.65	0.64	43
badminton_smash	0.62	0.6	0.61	40
squash_backhandboast	0.88	0.68	0.77	34
squash_forehandboast	0.69	0.83	0.75	35
Accuracy	0.7	0.69	0.69	152
SVM (C=1, gamma=0.0001, kernel='poly') GridSearchCV , PERCENTILE TRESHOLD=30				
badminton_clear	0.66	0.86	0.75	43
badminton_smash	0.82	0.57	0.68	40
squash_backhandboast	0.97	0.94	0.96	34
squash_forehandboast	0.97	0.94	0.97	35

Analiza RACKET

Numar total features	224
Numar total features variance treshold	217
Numar total features percentile 90%	201
Numar total features percentile 70%	157
Numar total features percentile 30%	67

Am ales sa fac analiza pe 3 variante de features pentru fiecare algoritm. Cu toate features, cu variance treshold aplicat, si cu percentile (30,70,90).

Am observat ca pentru SVM a contat mai mult un numar mai mare de features, dar totodata sa fie doar cele mai folositoare, de aceea cel mai bun rezultat a fost pe percentile 30.

Am observat ca pentru random forest, cel mai bun rezultat a fost pe toate features pe o adancime destul de mare.

Am observat la gradient boosting ca rezultatele sunt similare , mi-am dat seama insa tarziu, cand faceam analiza, ca diferentiatorul probabil statea in learning rate, si acolo puteam sa vad o diferenta concreta.

Ca analiza generala, am incercat sa testez pe cat mai multe varianta, astfel am vazut ca fiind foarte important sa folosesc GridSearchCV pentru ca o configuratie buna este foarte importanta. Am observat o diferenta clara intre o rulare de algoritm cu parametrii default si una cu parametrii optimi obtinuti prin Grid Search.

Cred ca este foarte important ca in functie de modelul/algoritmul folosit sa ne dam seama care este cea mai buna combinatie intre ce features oferim si ce parametrii. Pentru mine contra-intuitiv a fost ca pentru unele modele, un numar mare de features a fost mai facil vs. un numar mai mic de features mai importante, si vice-versa.

In alta ordine de idei, am observat si o corelare intre performantele algoritmilor si timpii de rulare.

Am observat ca in majoritatea cazurilor, loviturile de squash au fost cele mai bine clasificate. Am observat in plot bar ca sunt mai putine exemple pe cele 2 clase clasificate mai bine. (asta se poate vedea si in confusion matrix din ipynb)

Am observat ca loviturile de badminton sunt destul de asemanatoare

	Accuracy	0.86	0.84	0.84	152		Am observat ca loviturule de badminton sunt destul de asemanatoare, de asta clasificarile se incurca uneori, si din matricile de confuzie am observat ca se incurca intre ele.
	SVM(C=0.1, gamma=1, kernel='poly') GridSearchCV , PERCENTILE TRESHOLD=70						
	badminton_clear	0.67	0.6	0.63	43		
	badminton_smash	0.63	0.55	0.59	40		
	squash_backhandboast	0.79	0.91	0.85	34		
	squash_forehandboast	0.69	0.89	0.84	35		
	Accuracy	0.72	0.74	0.73	152		
	SVM(C=0.1, gamma=1, kernel='poly') GridSearchCV , PERCENTILE TRESHOLD=90						
	badminton_clear	0.55	0.63	0.59	43		
	badminton_smash	0.59	0.55	0.57	40		
	squash_backhandboast	0.82	0.68	0.74	34		
	squash_forehandboast	0.71	0.77	0.74	35		
	Accuracy	0.67	0.66	0.65	152		
	RandomForestClassifier DEFAULT PARAMETERS, ALL FEATURES						
	badminton_clear	0.7	0.81	0.75	43		
	badminton_smash	0.78	0.62	0.69	40		
	squash_backhandboast	0.94	1	0.97	34		
	squash_forehandboast	1	0.97	0.99	35		
	Accuracy	0.86	0.85	0.85	152		
	RandomForestClassifier (max_depth=8, max_features='log2', n_estimators=400) GridSearchCV, ALL FEATURES						
	badminton_clear	0.75	0.88	0.81	43		
	badminton_smash	0.87	0.68	0.76	40		
	squash_backhandboast	0.97	1	0.99	34		
	squash_forehandboast	1	1	1	35		

	Accuracy	**0.9**	0.89	**0.89**	152			
	RandomForestClassifier (criterion='entropy', max_depth=7, max_features='auto'), VARIANCE_TRESHOLD=0.25							
	badminton_clear	0.71	0.84	0.77	43			
	badminton_smash	0.78	0.62	0.69	40			
	squash_backhandboast	0.94	0.97	0.96	34			
	squash_forehandboast	1	0.97	0.99	35			
	Accuracy	0.86	0.85	0.85	152			
	RandomForestClassifier (criterion='entropy', max_depth=4, max_features='log2', n_estimators=300), PERCENTILE_TRESHOLD=30							
	badminton_clear	0.65	0.77	0.7	43			
	badminton_smash	0.74	0.57	0.65	40			
	squash_backhandboast	0.97	0.97	0.97	34			
	squash_forehandboast	0.97	1	0.99	35			
	Accuracy	0.83	0.83	0.83	152			
	RandomForestClassifier (criterion='entropy', max_depth=6, n_estimators=200), PERCENTILE_TRESHOLD=70							
	badminton_clear	0.73	0.81	0.77	43			
	badminton_smash	0.79	0.68	0.73	40			
	squash_backhandboast	1	1	1	34			
	squash_forehandboast	0.97	1	0.99	35			
	Accuracy	0.87	0.87	0.87	152			
	RandomForestClassifier (max_depth=8, max_features='log2', n_estimators=300), PERCENTILE_TRESHOLD=90							
	badminton_clear	0.73	0.84	0.78	43			
	badminton_smash	0.79	0.68	0.73	40			

	squash_backhandboast	0.97	0.97	0.97	34			
	squash_forehandboast	0.97	0.97	0.97	35			
	Accuracy	0.87	0.86	0.86	152			
	GradientBoostingClassifier, DEFAULT PARAMETERS, ALL FEATURES							
	badminton_clear	0.67	0.65	0.66	43			
	badminton_smash	0.68	0.65	0.67	40			
	squash_backhandboast	0.86	0.91	0.89	34			
	squash_forehandboast	0.94	0.97	0.96	35			
	Accuracy	0.79	0.8	0.79	152			
	GradientBoostingClassifier (max_depth=7, max_features='log2', n_estimators=200), ALL FEATURES							
	badminton_clear	0.77	0.84	0.8	43			
	badminton_smash	0.81	0.72	0.76	40			
	squash_backhandboast	0.94	1	0.97	34			
	squash_forehandboast	1	0.94	0.97	35			
	Accuracy	0.88	0.88	0.88	152			
	GradientBoostingClassifier (max_depth=7, max_features='log2', n_estimators=400), VARIANCE_TRESHOLD=0.25							
	badminton_clear	0.74	0.86	0.8	43			
	badminton_smash	0.84	0.68	0.75	40			
	squash_backhandboast	0.94	1	0.97	34			
	squash_forehandboast	1	0.97	0.99	35			
	Accuracy	0.88	0.88	0.88	152			
	GradientBoostingClassifier (max_depth=4, max_features='sqrt', n_estimators=300), PERCENTILE_TRESHOLD=30							

	badminton_clear	0.73	0.77	0.75	43			
	badminton_smash	0.76	0.72	0.74	40			
	squash_backhandboast	1	0.94	0.97	34			
	squash_forehandboast	0.95	1	0.97	35			
	Accuracy	0.86	0.86	0.86	152			
	GradientBoostingClassifier (max_depth=4, max_features='log2', n_estimators=300), PERCENTILE_TRESHOLD=70							
	badminton_clear	0.71	0.81	0.76	43			
	badminton_smash	0.79	0.65	0.71	40			
	squash_backhandboast	1	1	1	34			
	squash_forehandboast	0.97	1	0.99	35			
	Accuracy	0.87	0.87	0.86	152			
	GradientBoostingClassifier (max_depth=8, max_features='log2', n_estimators=500), PERCENTILE_TRESHOLD=90							
	badminton_clear	0.74	0.86	0.8	43			
	badminton_smash	0.85	0.7	0.77	40			
	squash_backhandboast	0.94	1	0.97	34			
	squash_forehandboast	1	0.94	0.97	35			
	Accuracy	0.88	0.88	0.88	152			