HyperParameter

1-VotingClassifier

Estimators: list of tuples like:

estimators=[('lr', log_clf), ('rf', rnd_clf), ('svc', svm_clf)]

voting: string:

voting='hard’

ده عن طريق predict class ثم بياخد ال Majority of vote.

voting='soft’

ده لما تكون ال models المستخدمة فى ال Voting بتقدر تتوقع ال probabilities ساعتها بدل ال Hard voting عن طريق ال Majority ، لا ده بيروح يشوف مجموع ال probability بتاعت كل class ويقسم على عدد ال models بعد كده يشوف لو كانت المجموع ده اكبر من ال threshold ولا اقل وبناء عليه يعمل assign ل class معين.

ده بيدى نتائج افضل.

2-BaggingClassifier

Base_estimator: object if left use DecisionTree()

انهى موديل عايز تستخدمه.

N_estimators: int

عدد ال models الى عايز تعملها run من ال model المستخدم .

Max_samples: int or float

عدد ال training instance الى عايز تعمل عليهم train فى كل estimator من ال n-estimators . او تعملها كنسبة بين 0 - 1 ك float number.

Max_features: int or float

كمان تقدر تتحكم فى عدد ال features المستخدمة زى عدد ال instance بظبط ، وكمان تقدر تحدد النسبة برضه .

bootstrap=True as default value

التغير من Bagging method ل Pasting عن طريق ال bootstrap الأساس هو Bagging لو هحول ل Pasting هخليها False ، الفرق بينهم فى ال Summary والى هو فى استخدام نفس ال subset ولا لا.

bootstrap_features=False as default value

زى bootstrap لو عايز كمان ال features تختارها random هتخليه ب True .

oob_score=False as default value

لما كل model يعمل run على جزء فقط من ال data فيه جزء تانى ال model مش هيشوفة خالص ، فهنا لو خليت ال oob_score=True بكدة هعمل على الجزء الى ال model مش هيشوفة Evaluation بدل ما اروح استخدم Croos-validation او اى طريقة لل Evaluation .

warm_start=False as default

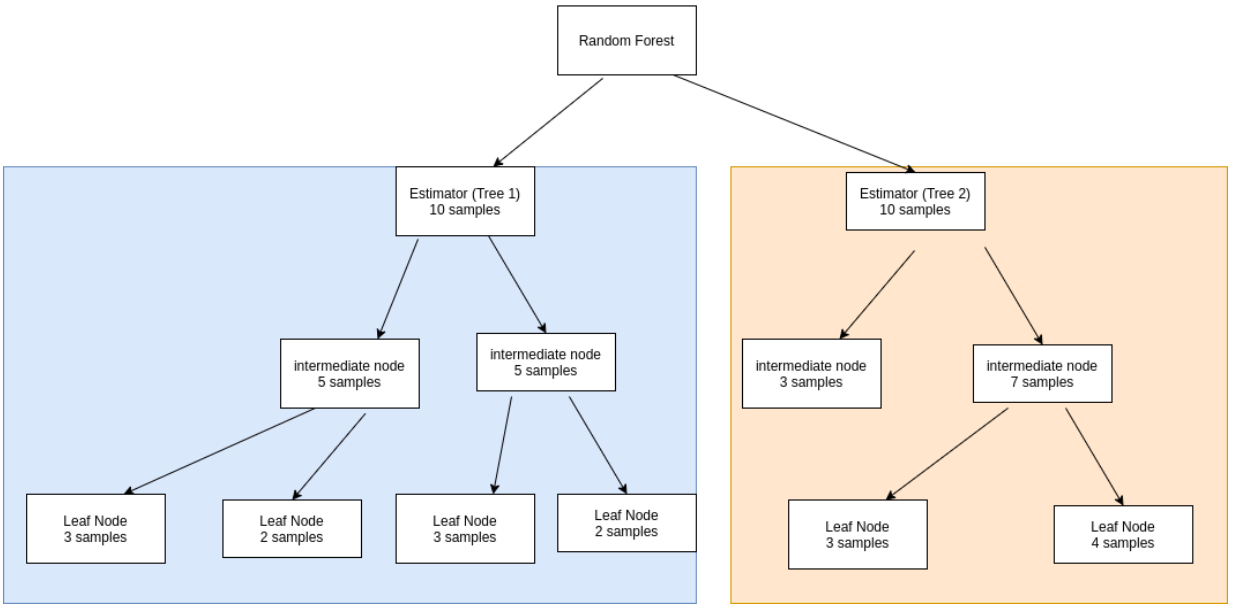
لو خليتها ب True هتستخدم اخر نتيجة وصلتلها من ال fitting الى عملته عشان تكمل عليها لما اجى اعمل run لل model من تانى ، لكن وهى False كده فى كل مرة بعمل fit بيعمل training من اول وجديد .

n_jobs=int

لو خليتها n_jobs=-1 هتعمل run باستخدام كل ال CPUs المتاحة ، فهى بتخلينى ققدر اعمل run على التوازى ، وده لان كل model اصلا منفصل عن التانى .

3-RandomForest

N_estimators = 2
Max_depth = 2 (look at estimator not Random Forest node)
Min_samples_split: *4*
min_samples_leaf: 2
max_leaf_nodes:*4*



هو عبارة عن ensemble من ال Decision Tree عشان تستخدمه علطول ، فهو عنده معظم ال hyperparam بتاعت Decision Tree بلس ال HyperParam الخاصة بال Ensemble method .

N_estimators: int

عدد ال models الى عايز تعملها run من ال model المستخدم ، ده خاص بال Ensemble method .

criterion{“gini”, “entropy”}, default=”gini”

انهى نوع من ال loss function محتاج تطبقه من الموديل ده خاص بال Decision Tree.

Max_depth:int, None as default

طول ال Tree فى كل estimator آخرها اد ايه وده خاص بال Decision Tree .

Min_samples_split*int*:int or float default=2

اقل عدد من ال instance يكون موجود فى ال Node عشان يحصل split وده خاص بال Decision Tree .

min_samples_leaf:*int or float, default=1*

أقل عدد من instances يكون موجود فى ال leaf node وده خاص بال Decision Tree .

max_leaf_nodes:int, default=None

اقصى عدد من leaf nodes ممكن يكون موجود فى كل Tree وده خاص بال Decision Tree .

باقى ال Params معظمها زى الى فى *BaggingClassifier* .

4-ADABOOST

Base_estimator: object if left use DecisionTree()

انهى موديل عايز تستخدمه وعلى حسب بقا ال hyper param بتاعت ال model نفسه انتا مظبطها ازاى هو هيعمل Train بيها .

N_estimators: int

عدد ال models الى عايز تعملها run من ال model المستخدم .

Learning_rate: float, default=1.0

ال learning rate يتناسب عكسي مع عدد ال estimator عشان تقدر تتحكم فى ال regularization.

algorithm{‘SAMME’, ‘SAMME.R’}, default=’SAMME.R’

ال default شغال بال probability بينما الاخر بيجيب ال pediction علطول .

4-Gradient Boost

نفس موضوع ال Random Forest برضه خاص بال Decision Tree بس بدل ما بيرن على التوازى زى ال Random Forest لا بيرن بطريقة Sequential زى ما وضحنا فى البوستات والملخص.
نفس ال HyperParams فى الى فاتوا .

تَمّ بِحَمْدِ اللَّهِ .