Summary

Parallel Methods

Voting Classiers

ال Voting classifier هو طريقة من خلالها بنعمل run لأكتر من model على التوازى (differents models) ، وفى الاخر بناخد ال prediction بتاع كل model وناخد اكتر class كانت ال models ديه بتقول عليه بمعنى :

Model 1 say class 0
Model 2 say class 1
Model 3 say class 0

يبقا كده ال Majority of vote is class 0 .

بيرن على موديلز مختلفة .

Bagging and Pasting Classifier

بدل ما اعمل run ل models مختلفة ققدر هنا اعمل run لنفس ال model اكتر من مرة والاساس هنا هو ال DecisoinTree لكن ققدر استخدم models تانية. وهنا كل model بي run على subset من الداتا مع تغير ال subset ده فى ال bootstrap aggregating which is Bagging ، او على نفس ال subset والى هو pasting .
الميزة فى ال Bagging انه بيعمل Soft voting لو كان الموديل ليه probabilites .

بيرن على نفس الموديل فقط ولكن تقدر تستخدم اى موديل.

Out-of-Bag Evaluation

فى ال Bagging بيحصل انى ال models بما انها بتعمل run على subset مختلفة من ال data ، هنا فى بعض ال models مش بتشوف باقى ال training data ، وهنا ققدر استخدم الجزء ده فى انى اعمل عليه evaluation بدل ما استخدم cross-val method.

Random Patches and Random Subspaces

ال Random patches لما اخلى ال instances وال features يختارو random عن طريق bootstrap=True & bootstrap_features=True ، و Random_subspaces لما يكون فقط ال Randomness ديه فى ال features فقط وهنا bootstrap_features=True & bootstrap=False .

Random Forests

بدل من استخدام ensemble method عن طريق Baggin or pasting وتعمل pass ل decision Tree ال Random Forest هو implementation ل Ensemble method باستخدام Decision Tree ، وفى معظم ال param بتاعت ال ensemble + Decision Tree ، ال Random Forest اسرع من انك تعمل run ل Baggin or Pasting Classifer باستخدام ال Decision Tree ، لانها بتختار احسن feature تعمل عنده split لل node من خلال random subset من ال features مش كل ال features .

Extra Random Forests

بدل ما تختار best feature لا بتختار random threshold for each feature تعمل من عنده split لل node وده بيكون اسرع بكتير .

Feature Importance

فيه ميزة قوية جدا فى ال Random Forest انك تقدر تحصل على اهم features بعد ما تعمل train عن طريق ده تقدر تعمل select ليهم وتعمل traing من تانى وديه طريقة كويسة جدا لل Features Selection .

كل الى فات كنت تقدر تشغله على CPUs or server مختلفة على التوازى .

Sequential Methods

Boosting

الطرق الى فاتت كان عن طريق انك بتعمل run لل models بس كل model على حدا وفى الاخر بتاخد بال Voting.
هنا هبتدى انى اعمل run sequential for models ، كل واحد بيحاول انه يعمل correct للي قبله .

AdaBoost (Adaptive Boosting)

واحد من الطرق ديه هو AdaBoost ، وهو انى بروح اعمل run لاول موديل ، وبعد اعمل evaluate ليه على ال data الى عمل عليها train ، واشوف انهى instance حصل فيها Wrong classificatoin ، واروح اعمل updates لل weights المتعلقة بال instances ديه ، ومن ثم اروح اعمل run للموديل اللي بعديه ، وهكذا لحد ما اوصل للعدد ال انا محدده من ال models .

هنا بيحصل انه يعمل weights لكل instance على حد ونفس الكلام كل estimator بيكون عنده weights مختلفة .

بيرن على نفس الموديل فقط ولكن تقدر تستخدم اى موديل.

Gradient Boosting

بدل ما اروح اعمل update لل weights بتاعت كل instance على حد ، لا هنا بروح اعمل updates لل weights بناء على ال error المتبقى ، بمعنى كل model بيروح يعمل predict على ال data الى عمل عليها traing ، ثم ببتدى اشوف الفرق بين ال y_actually - y_predicted ، ولما اروح اعمل fit للموديل الجديد اعمل passing للفرق ده .

```
tree_reg1 = DecisionTreeRegressor(max_depth=2)
tree_reg1.fit(X, y)
y2 = y - tree_reg1.predict(X)
tree_reg2 = DecisionTreeRegressor(max_depth=2)
tree_reg2.fit(X, y2)
```

 # Note we pass y2 the residual errors made by the previous predictor.

بيرن على نفس الموديل فقط وفقط مع ال Decision Tree .

تَمّ بِحَمْدِ اللَّهِ .