Summary

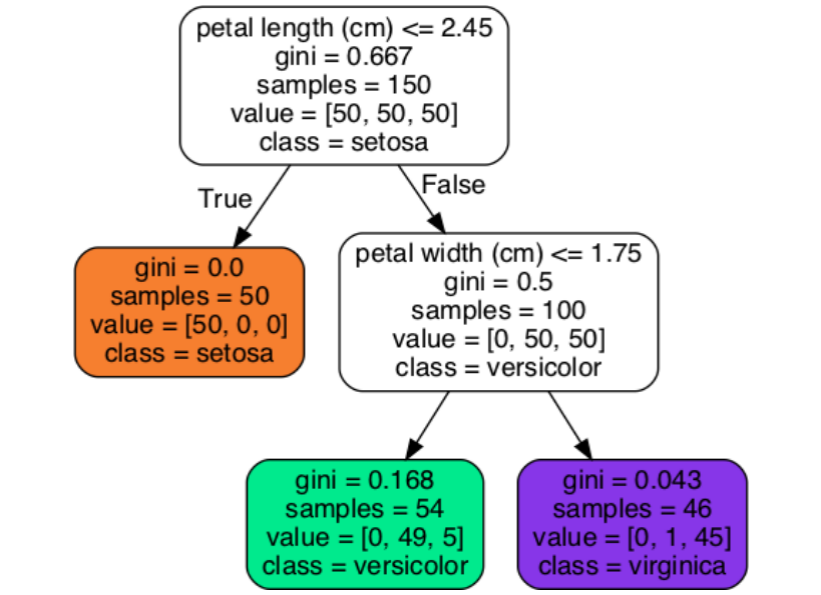
Decision Tree

هو machine learning model نقدر من خلاله برضه نحل بيه classification, regression ل complex data

Training and Visualizing a Decision Tree

بنعمله Train عادى جدا ولكن نقدر نعمل visualize لل Tree ديه ، بعد ال Training عن طريق استخدام export_graphviz ، ديه method موجود فى ال Sklearn برضه.

زى ما فى الصورة.



الموضوع ببساطة روح شوف احسن feature نعمل من عنده split لقى انه petal length ، بعد كده حدد threshold لل feature ده لقى انه 2.45 ، بعدين روح قسم ال data على اساسه ، كل ال instances الاكبر من 2.45 هتروح يمين ، فنلاقى هنا فيه Node تانية حاصل منها split ، يبدء يشوف feature غير الى خده ، ويعمل نفس الكلام.

ال gini ده بيقولك ال impurity بتاعت ال Node فمثلا الى فيها gini=0 ديه pure خالص لان مفهاش غير كلاس واحد.

بينما الى فيها 0.168 ، فيها classes مختلفة حسبها ازاى بقا :

1 - (0/54)^2 - (49/54)^2 - (5/54)^2

ال 54 ده هو عدد ال sampels فى ال Node.

Making Predictions & Estimating Class Probabilities

طب ازاى هيتوقع instance جديد هيروح يمشى فى ال Tree ويشوف، هل هو اكبر ولا اصغر من ال petal length بناء عليه يعرف هيمشى فى انهى طريق ، لو كان اقل يبقا خلاص هيتصنف ك Setosa ، ولو كان ال node فيها اكتر من class زى left leaf node of max depth 2 الى فيها gini=0.168 ، هيروح يتوقعها بال class الى واخد highest probability . لكن لو طبعت ال probability هتلاقى مديك النسبة
[0/54, 49/54, 5/54] .
لانها 5 اكبر من 2.45 ولان 1.5 اصغر من 1.75 راح ل node الى على الشمال فى depth 2 الى فيها gini=.0168 ، وبما ان عندنا Three classes فلما يعمل predict_prob يديلة percentage of each class this instance may belong to .

```
>>> tree_clf.predict_proba([[5, 1.5]])
array([[0.        , 0.90740741, 0.09259259]])
```

The CART Training Algorithm

هو ال Classification And Regression Training Algorithm ببساطة شوف احسن feature واحسن threshold ينفع نعمل split بناء عليه والى بيكون بيقلل impurity قدر الإمكان ، لقيته اعمل split into 2 node ، لان sklearn كل ال trees الى فيها هى **Binary Tree** ، او بناء على ال another loss function والى هى ال Entropy ، ونفس الكلام بيحصل من عند ال Nodes الجديدة وهكذا.

طب امتا بيقف بناء على ال Hyper Parter الى انتا بتحددها سواء قلتله اخرك توصل لعدد معين من ال Leaf nodes ، او انك مينفعش تعمل split لو كان ال node فيها عدد معين من ال Samples وهكذا ، ال hyper param معمول ليها ملخص برضه ، كل واحد بيعمل ايه .

Computational Complexity

لو شوفنا فوق ازاى بيحصل prediction هنلاقى انى الموضوع بسيط هو بيدى ال instance لل tree فيمشى فيها وبناء على كل node بيروح شمال او يمين لحد ما يوصل ل leaf node وهنا ده فقط عباره عن طول ال Tree :

O(log(m))

لكن ال Training complexity بياخد وقت كبير جدا لانه عباره عن انه بيمشى فى ال Tree ده كده:

O(log(m))

ويمسك كل instance فى الداتا ده عباره عن:

O(instances)

ويروح يشوف كل feature جوه ال instance ده هل هو افضل واحد اعمل من عنده split ولا لا وده:

O(n)

الناتج هو:

O(instances*n*log(m))

Regularization Hyperparameters

لو فيه مشكله Over Fitting او under fitting انتا تقدر تعمل Regularization لل Tree بتاعتك عن طريق ال Hyper Parameter الخاصة بيها سواء انك تتحكم فى طول ال Tree او انى يكون اخرها عدد معين من ال Leaf nodes وغيره ، وده متكلمين عليه اكتر فى ملخص عن ال Hyper parameters .

تَمّ بِحَمْدِ اللَّهِ .