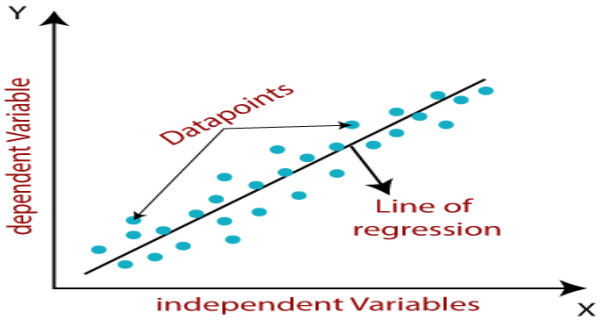
Machine Learning algorithms

: Regression <=

Linear regression ☐

: Linear regression<=

☐ فكرته اتشرحت في سيشن ال Calculus ارجع بص عليه هناك 😀



: Classification<=

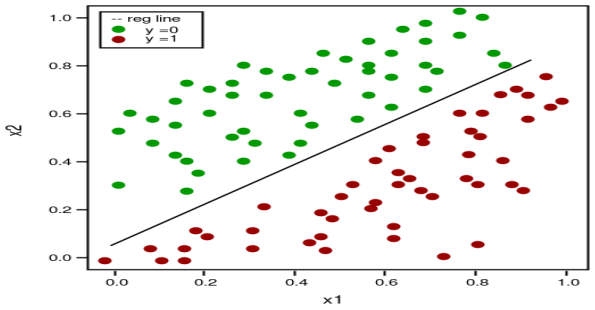
Logistic regression ☐

SVM ☐

Decision tree ☐

KNN ☐

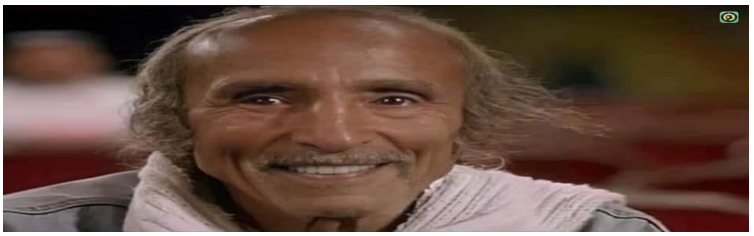
Logistic regression<= :



- ☐ ي ال logistic انا بعمل equation زي بتاعت ال linear بالظبط بس الفرق هنا اني مش ه fit الداتا عليها انا هشوف الي قبلها والي بعدها
- ☐ طب الي قبلها والي بعدها ده بيتحدد على اي اساس ؟
- ☐ بيتحدد على اساس ال activation function ودي شغاله ازاي هنا ؟
- ☐ لو معايا binary classification بتطلعلي النواتج ما بين ال 0 و 1
- ☐ لو الناتج اقل من0.5 يبقى 0 لو اكبر يبقى 1

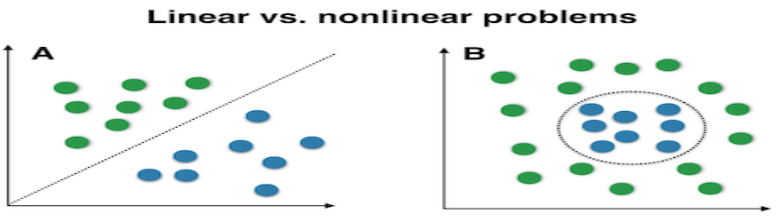
=> يبقى ال logistic regression و ال linear regression بيفترضوا إن العلاقه خطية بس الفرق إن ال logistic الخط هنا as boundary بيفرق ما بين الداتا

=>الكلام الي فوق ده جميل في حالة وجود classes 2 واحد بس طب لو انا معايا classes 3 ؟



- ☐ يبقى هرسم 2 boundary عشان افرق ما بينهم
- ☐ هنا الموضوع بيصعب بس كل اما الفروقات ما بين الداتا وبعضها تكون كبيره كل اما ال boundary دي هيكون سهل اننا نرسمها
- ☐ الكلام ده كله في حالة ال Linear .
- ☐ لو عندنا 2 classes بقى في حالة ال Non-linear ؟
- ☐ مش هعرف اعمل ما بينهم boundary عشان هت fit على الداتا الي معايا بس فهنستخدم ال SVM .

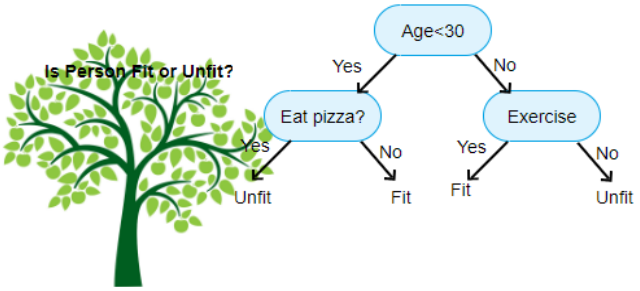
: SVM<=



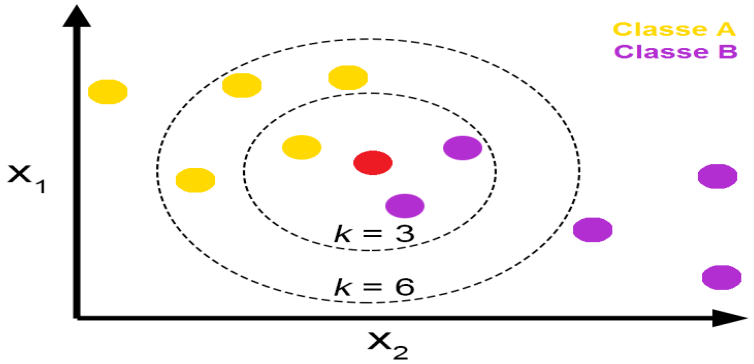
- ☐ الرسمه شارحه نفسها فمش هشرح 😁

: decision tree ال <=

- ☐ هي مجموعة if-else على هيئة tree .
- ☐ يعني هي عبارة عن شوية conditions والـ conditions دي بتتحط في ال tree على حسب ال priority بتاعتها.
- ☐ بتشتغل على داتا categorical .
- ☐ بتستخدم في ال classification والـ regression لكن استخدمها الشهير في ال classification .



: KNN ال <=



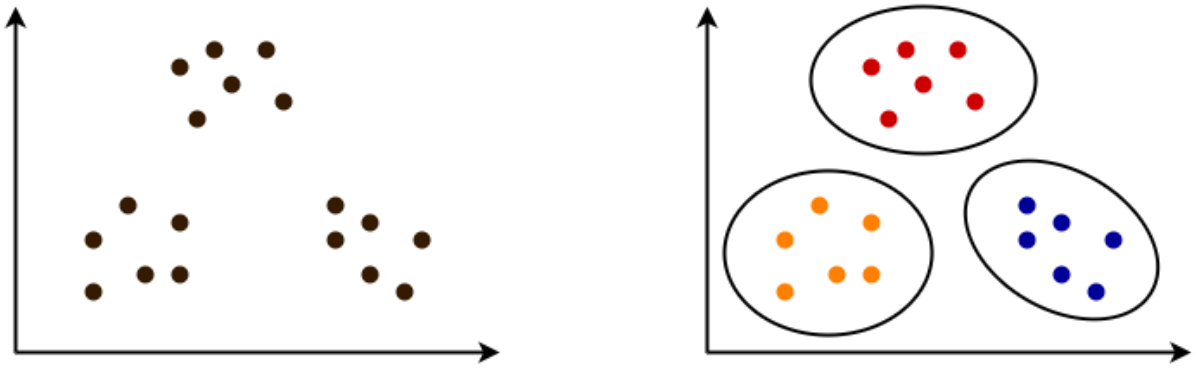
- ☐ النقطة الحمرا الي في النص دي اسمها query point
- ☐ على حسب ال k الي هي عدد ال neighbours هيجبلك نفس العدد من الداتا او الpoints القريبه من ال query point

- ☐ يعني اول k=3 هيجيب اقرب 3 نقط من ال query point والي هما عباره عن 2 من class B وواحده من Class A
- ☐ ولو انت اديت للمودل ده داتا جديده عشان يعملها classification فهو هيستخدم ال voting يعني هيشوف اكبر عدد عنده في الداتا من class A or Class B
- ☐ في المثال بتاعنا ده الداتا اغلبها في class B فهي classify الداتا الجديده في class B

: Clustering<=

- ☐ K-Means
- ☐ Hierarchical clustering

: K-means<=



Before K-Means

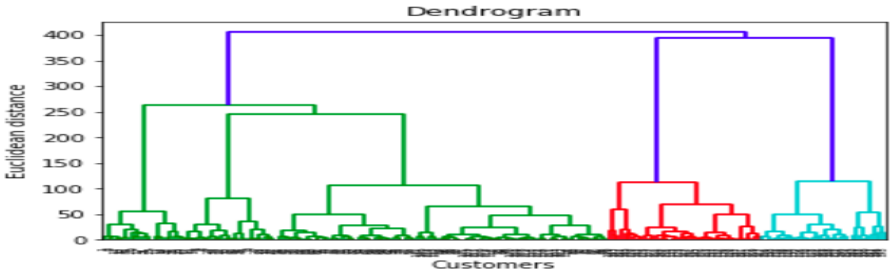
After K-Means

- ☐ مش هشرح برضو

: Hierarchical clustering<=

- ☐ هو في البداية بيعتبر كل point في الداتا بتاعتي هي عباره عن cluster لوحده
- ☐ وبعد كده بيبداءمثلا ياخد كل 2 clusters قريبين من بعض عشان يعملهم cluster واحد
- ☐ وبعده كده كل 3 clusters وهكذا لحد ما كل الداتا تبقى cluster واحد

- ال output بتاعي اسمه dendogram



: Random forest algorithm <=

- هو عباره عن updated algorithm من ال decision tree
- طب ايه المشكله الي خليته يظهر ؟
- وليكن مثلا في قاعده ثابته ان معظم الناس الي بتروح mac بيطلبوا طلب معين ويبقى انا ه predict ان اي شخص هيروح mac هيطلب الطلب الفلاني صح ! صح
- يعني دي قاعده ثابته مفيهاش تعديل
- طب افرض ان 50% من الناس راحوا MAC يفطروا والباقي يتغدى
- يبقى انا كده مينفعش اعمم قاعده عامه على الداتا الي معايا دي
- يبقى المشكله في ال decision tree ان لو الداتا بتاعتي مفيهاش rules عامه وواضحه وثابته مش هينفع استخدمه
- والمشكله كمان لو الداتا unbalanced يعني مثلا لو جيت جمعت الداتا قدرت اجمع 60% من الناس الي بيروحوا mac يفطروا و40% بيتغدوا هل ده معناه ان اغلب الناس بتروح mac عشان تفطر ؟
- لا ده معناه إن ال data set بتاعتي اغلب الناس الي فيها بيفطروا فالداتا مش balanced وده نتيجة إن احنا مقدرناش نجمع داتا كفايه
- الحل بقى اننا نروح لل random forest

- ☐ مبدائيا ال random forest وال decision tree بيستخدموا في الداتا الي ال values بتاعتها categorical زي male/femel, yes or no يعني قيم معينه وثابته

=>ال Random forsest بيشتغل ازاي ؟

- ☐ هو فكرة اني بعمل كذا decision tree وكل واحده منهم بديها random data
- ☐ فده بيخلي كل decision tree تتدرب على داتا مختلفه
- ☐ فكل واحده بتطلع نواتج مختلفه مثلا لو معايا 10 decision tree في منهم 7 اجمعوا على حاجه معينه و 3 اجمعوا على حاجه مختلفه فده معناه ان ال 7 يكسبوا
- ☐ بس الفكره لو 5 قالوا حاجه وال 5 الثانيين قالوا حاجه تانيه يبقى كده معملناش حاجه
- ☐ والأوحش لو 6 قالوا حاجه و 4 قالوا حاجه تانيه
- ☐ فعشان كده بنحاول نزود عدد ال decision tree عشان ميحصلش حاجه زي كده
- ☐ لما ينيجي نحكم على ال output بتاع ال algorithm ده بنديله test data لو جابها كلها صح يبقى تمام
- ☐ وال algorithm ده من ال algorithms الي بت overkill
- ☐ يعني المودل ده بيجيب accuracy عاليه جدا
- ☐ لإن احتمال ال overfiting الي فيه عاليه جدا لإنه بياخد ال rules بتاعته من الداتا الي هو متدرب بيها
- ☐ فاحنا هنا بنهتم جدا بالداتا بتاعتي وطريقه تجميعها عامله ازاي وال domain knowledge
- ☐ يعني المودل ده بي generalize على الداتا بتاعته بس فعشان كده لازم ناخد بالنا من الداتا

=> ال Cross Validation :

- ☐ ايه سبب ظهوره ؟
- ☐ نفترض ان معايا داتا بالشكل ده "1,2,3,4,5,6,7,8,9"

- فانا مثلا عاوزه ادرب المودل على "1,2,3,4,5,6" واخليه ي test على "7,8,9"
- الفكره إن الداتا هنا حجمها قليل اوي لدرجة انه "7,8,9" ميعرفش عنهم حاجه خالص فهو صعب اوي يستنتجهم من خلال شوية الداتا الي اتدرب عليهم
- طب احنا هنفترض انه عرف يجيب ال test data صح ال accuracy حرفيا هتبقى في الأرض فمينفعش اعتمد عليها
- الكلام ده بيحصل لو الداتا قليله او الداتا كبيره عاديه بس مفيش تشابه او ال correlation مابين الداتا وبعضها قليله جدا
- ففكرة ال cross validation عامله زي فكرة لما انا وانت واطفال مصر كلهم في فترة الأعداديه كنا عيال دحيحه كنا بنذاكر ونحل كل المحافظات عشان نحاول على قدر الإمكان نبقى متأكدين إن كل اسئلة الامتحان في جيبنا , فهو بيعمل كده بالظبط
- مره هياخد "1,2,3,4,5" يتدرب بيهم وي test ب "6,7,8,9" فمش هيعرفهم وهيسقط
- هيتدرب مره كمان ب "1,2,3,6,7,8" وي test بالباقي هيعرف شويه وشويه لا وكمان مش هيبقى متأكد
- هيفضل يعمل الكلام ده لحد ما حرفيا يذاكر كل الداتا وعمل test بكل الداتا برضو
- بس نخلي بالنا انه كده بي overfit على الداتا دي بس

=>امتى استخدم ال cross validation :

- ☐ لو الداتا قليله ومش هعرف ازودها
- ☐ لو الداتا كثيرة الاحسن اننا نغير ال model احسن ما نعمل cross validation بعد كده نغير ال hyperparameters

=>دي مش كل ال algorithms الي في ال ML بس نقدر نقول دول اشهر حاجه بتستخدم واخدنا تاسكات جيبنا فيهم algorithms تانيه كثير ارجع بص عليها .

اتمنى اكون افدتكم فضلا ادعوا لأمي بالشفاء 🤲🏻