

## Task 2 of ML

### Gradient Descent

مثلا لو عندك دالة فيها خطأ وانت عايز تقلله، الحل إنك تبدأ من أي قيمة عشوائية، وتمشي خطوة خطوة في الاتجاه اللي يقلل الخطأ. كل ما يبص تحت يلاقي الميول فين وينزل ناحية العكس بتاعه.

وال **learning rate** هو سرعة الخطوات. لو كبيرة أوي ممكن تنط من مكان لمكان وتضيع، ولو صغيرة أوي هتفضل تمشي ببطء وممكن تزهدق ، ف بنعملها وسط بحيث انا توصل للمكان اللي انا عايزه وفي الاغلب بيكون عند الميل بصفر و اللي بيتعمل بالموضوع ده كثير، زي في الانحدار الخطي واللوجستي وكمان الشبكات العصبية. وفيه أنواعه: ممكن تحسب على الداتا كلها مرة واحدة (ده بطيء شوية)، أو تاخذ عينات صغيرة (ده الأسرع غالبًا).

المشاكل اللي بتحصل إنك ممكن تختار learning rate غلط.

### ROC Curve

الفكرة وما فيها ان لو عندك موديل بيطلعك احتمالات (يعني يقولك الشخص ده عنده مرض بنسبة 0.7)، فأنت بتختار threshold: فوق كذا يبقى مريض، تحت كذا مش مريض. وهنا بيجي الـ ROC curve. بيوريك العلاقة بين الحساسية (قد إيه جبت الناس المرضى صح) وبين الإنذارات الكاذبة قد إيه قلت على ناس أصحاء إنهم مرضى كل ما تغيّر threshold يطلعك نقطة جديدة على المنحنى.

وفي الآخر بنبص على حاجة اسمها AUC ، يعني المساحة تحت المنحنى. لو 1 يبقى الموديل ممتاز، لو 0.5 يبقى زي عشوائي وغير مضبوط : يعني مثلاً مرض زي السكر ، ممكن تقبل إنك تقول على بعض الأصحاء إنهم مرضى (خطأ بسيط) بس مقابل إنك ماتقوتوش مريض حقيقي ، لان ده ممكن يسبب مشكلة للشخص المريض والحالة بتاعته تدهور .

### KNN

K-nearest neighbors هو ان يكون عندك بيانات متخزنة، وتدخل حاجة قريبه من حاجة في البيانات اللي انت خزنتها و تبص على أقرب جيران ليه وهيكون عدد الجيران اللي هتبص عليهم K وتشوف أكثر فئة فيهم إيه وتقول هو زيهم.

لو اخترت K صغير قوي، ممكن جار غريب بيوظلك النتيجة. لو كبير جدًا، هتضيع التفاصيل. فبتجرب أرقام زي TD HG,S' 3 أو 5 أو 7 وتشوف إيه الأنسب.

مشكلة كمان إنك لازم تطبط البيانات (scaling) لأن لو عندك خاصية كبيرة في القيم هتسيطر على المسافة كلها.

الميزة إنه سهل، بس مشكلته انه ثقيل وبطيء لما الداتا تبقى كبيرة.

بيستخدموه في حاجات بسيطة زي تصنيف صور صغيرة أو أنظمة توصية خفيفة.