Tasks

=>ايه هي مميزات ال super harvard modified ؟

- الي عملوه زياده عن ال modified انهم كبروا مساحة ال cache memory
- حطوا I/O controller

=> امتى نستخدم pip ومنستخدمش conda ؟

- لما يكون في library مش موجوده في Conda وموجوده في pip لكن في العموم conda احسن من pip لأنها وعلى الرغم انها أبطاء فهي بتوفر كل ال dependencies الي بتعتمد عليها ال library

- بس عيب conda انك لو نزلت حاجه ب version وجيت تنزلها ب version تاني فهي هتنزل ال version 2 اما pip هيعمل override

=>ايه هي ال solid principles ؟

1-single responsibility وهو انه ال class يكون مسئول عن حاجه واحده بس وبالتالي لما يجي يتغير فهو هيتغير عشان سبب واحد بس

2-open closed principleوهو ان ال classes وال moduls وال functions لازم يكونوا open لل extension و closed لل modification يعني اقدر اعمل add لأى behaviour انا عاوزه اضيفه عليهم من غير ما اغير ال behaviour الاساسي بتاعهم

Liskov substituations-3 هو انه الابن يقدر يحل محل الأب لو غاب بمعنى عندنا class T وclass S وهو child ل T فمعنى كده انه S يقدر يعمل اي حاجه بيعملها T فبالتالى لو جيت عملت كده() T t=new T ده الطبيعي لكن كمان ممكن تعمل كده

```
T t = new S()
```

Interface segregation-4 وهو ببساطه اني بقسم ال Interfaces الكبيره ل Interfaces صغيره عشان لو جينا نعمل لحاجه منهم implement ناخذ بس ال methods الي محتاجينها

Dependency inversion-5 ان لو في high module بيعتمد على low module منخليهمش يعتمدوا على بعض عن طريق ال abstraction

=>ايه هو ال distributed system topology ؟

- ال topology بتاعت ال ds يعني ازاي الأجهازة دي متوصله ببعض او ايه هي طرق توصيل ال network مابينهم

- بنحدد اي topology على اي اساس ؟ على اساس ال system بتاعنا يعني في مثلا system عاوز سيرفر يبعت لكل الاجهزة ومفيش حاجه تتبعتله و system تاني محتاج انه السيرفر يبعت ويتبعتله وهكذا

=> ايه هو dockerization, containerizations , kubernetes, Jenkins؟ "الحوار كبير فكوبايتك وتعالى "

- خلونا نجيب الشريط من اوله ونتكلم ليه الحاجات دي ظهرت كان في مشكله زمان انه اغلب المشاريع الضخمه بتكون عباره عن مجموعه كبيره من ال tools وال pacages وال libraries المعتمده على بعض والكود بتاع

الprojectبيكون معتمد عليها بشكل اساسي وده بيدخلنا في حاجه اسمها ال dependancy hell

- المشكله اننا لما نيجي ننقل ال app بتاعنا من environment لمكان تاني خالص بتضطر انك ت install كل ال packages وال libraries دي تاني يعني مثلا خلصنا ال app بتعنا على machine شغاله windows مثلا فانت عاوز تجرب الappعلى linux فهتضطر انك تسطب كل الحاجات الي ال app ده بيعتمد عليها على الجهاز الي فيه linux

- ال containerizations ك consept فكرته انه عندنا حاجه اسمها container بنجمع فيه كل ال packages وال libraries الي ال app بتاعنا هيحتاجها وهيوفرله ال enviroment كمان المناسبه ليه عشان ي run يعني يوم ما تحب انك تجرب ال app بتاعك في operating system تاني او اي جهاز تاني عموما ال container بيقولك ملكش دعوه اديني انت ال app وانا هعمل كل حاجه تخليه ي run

- والي بيطبق الفكره دي هو ال docker فمفهوم ال dockerization ما هو الا اننا بنطبق مفهوم ال containerizations باستخدام برنامج ال docker وهوعباره عن open source cloud بترفع عليه ال app بتاعك وتrun عليه

- في مفهوم تاني ظهر قبل الcontainerizations وهو ال virtualizations وده فكرة اني انزل os تاني على نفس ال operating system الي انا شغاله عليه as virtual machine بدل ما يجبوا جهاز تاني ب os مختلف لا هما هينزلوا وهيجربوا عليه ال app بس مشكلته انه بيستهلك ال RAM وال process بتبقى بطيئه جدا فال container ظهر بعد الفكره دي عشان يقدروا يقللوا من استهلاك ال ram وال processes متتأثرش

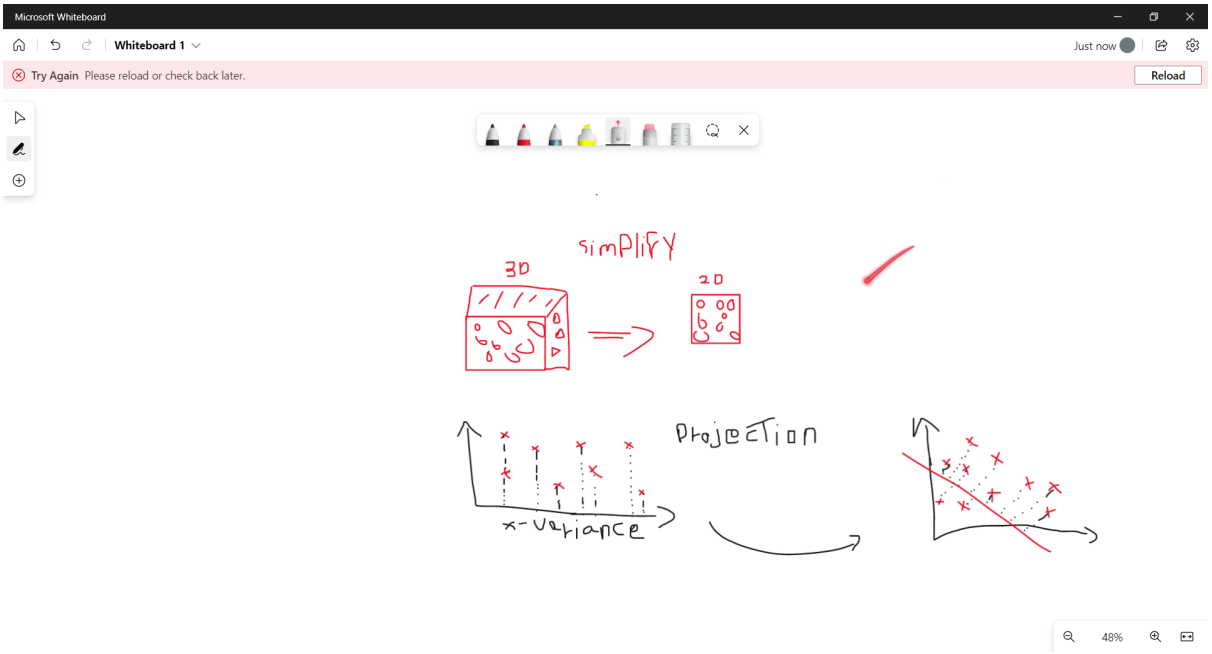
- خلصنا docker "قول الحمد لله " ندخل على kubernetes "قولتلك الحوار محتاج خربوش شاي "

- الفكره بتاعته جات منين ؟ هي جات لما جوجل كانت محتاجه تنظم ال containers بتاعتها وتجمعهم تحت سيطرة حاجه بتنظمهم وتتحكم فيهم كلهم وعملوا حاجه اسمها BORG بتعمل كده وبعدها ه عملوا حاجه تاني افضل اسمها omega وبعدين جوجل قررت ت share الكلام ده مع العالم كله "اكيد

مش لله والوطن " فاخترعواkubernetes الي هو عباره عن Container كبير في ال containers صغيره وال containers دي في حاجات مشتركه فيها فعشان كده اتحطوا في نفس ال container الكبير

- ندخل على Jenkins اخر حاجه معانا "تعبت ؟ قومت كملته للي قرأها صح عاش ي سنجل يا بائس الناس التانيه ربنا يغفرلكم "
- هي عباره عن open source مكتوب ب java بنستخدمها في عملية ال continuous integration وال continuous delivery لما تيجي ترفع الكود بتاعك على github اول ما بتعمله push بياخده يعمله build وبي test الكود لو في اي خطاء فهو هيوقف عملية ال building وهيلغي الحاجه الي انت رافعته لحد ما تظبط الكود بتاعك ولما تصححه ويعدي من مرحلة ال test هيعمله production

=>ايه هو ال PCA :



- هو فكرة ال principal component analysis وهو لو انا معايا كذا dimension بحاول اني اقللها ولكن عشان ده يحصل في شوية مشاكل هتقبلنا فاحنا بنحاول ان ال algorithm او الmechanism الي شغال بيه ال PCA يتفادى هذا المشاكل
- اول حاجه بعملها اني ب normalize الداتا
- ثاني حاجه فكرة اني اضغط الأبعاد فانا بضحي بأهمية كل بُعد مش كليا انا بضحي ببعض ال features الي في كل بُعد

- بس الفكرة اننا بنحسب حاجه الأول اسمها ال correlation وهو اني انا اشوف تأثير كل ال features في تغيير الناتج يعني هل في features معتمدين على بعض بشكل او بأخر , ممكن يكون في features ملهاش لازمة , وممكن يكون في features لازم أبقى محتفظ بيها
- الكلام ده عشان تعمله لازم تبقى mathematician قوي

=>ايه هو ال data unicorn ؟

- هو شخص بيعرف يعمل كل حاجه عنده domain knowledge فاهم data structure فاهم software engineering فاهم algorithms فاهم ML يعني من الأخر بيعرف يعمل كل حاجه .

=>ايه هو ال OOD or out of domain ؟

- نه انا مثلا اكون مدرب المودل على قطة وكلب وحصان فانا اروح مديله صورة زرافه فيقولي انه ood يعني ده ولا حاجه من ال classes الي قولتلي عليها .

- طب ده بيحصل ازاي ؟

- الفكره كلها ان القطة والكلب والحصان والزرافه كل ال classes دي ليها patterns معينه فهو كل الفكره انه بي check على ال patterns دي هل الزرافه تنتمي لأي pattern من الي عندي بنسبة معينه الي احنا بنقول عليها threshold.

=>ايه هو ال unsupervised regression ؟

- هو فكرة اننا ن train الداتا من غير وجود labels ليها يعني من غير وجود نواتج صحيحه ليها على أمل اننا نلاقيها فبقيس ال similarity

- Concept ال regression هو اني بيبقى معايا الإجابات الصحيحه بس ال output بتاعي بيطلع في continuous range

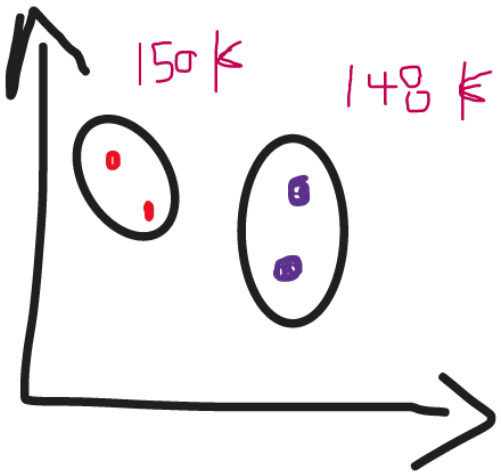
- يبقى في ال unsupervised انت معكش القيمه الصحيحه بس بتقيس ال similarity

- في ال regression انت معاك القيمه الصحيحه بس في continuous range فبتقيس البُعد

- فكده ال unsupervised regression النواتج هتطلع في continuous range بس مش النواتج او القيم الصحيحه

- لما بنيجي ن cluster كنا بن cluster على اساس similarity او ال distance فهو احنا هنا نفس الفكره اننا بنحاول نلم الداتا الي ملهاش labels دي وهنحاول اننا نقسمها ل groups عشان نشوف الداتا القريبه من بعض في قيم ال features

- وبعد كده بنفترض قيمه افتراضيه لل data دي وبناء على ال similarity الي ما بينها وما بين بعض هتبداء تقلل او تزود
- مش فاهم صح ؟



- يعني ولنفترض إن دول شقق والنقطتين الحمر دول قريبين من بعض في ال features فقولت انهم هيتبعوا نفس الطريقة في تسعير الشقه وهكذا بالنسبه للنقطين التانيين

- وروحت افترضت قيمة لسعر الشقه في بالنسبه للنقط الحمرة انهم ب 150 الف والنقط البنفسجي ب 148 الف لأنها مش بعيده اوب عن الجروب التاني

- يعني انا هنا هبداء اوزع الاسعار بناء على ال distance مابين الشقق وبعضها
- طب هتيجي تقولي طب وانت ايش عرفك انها ب 150 الف
- هقولك احنا هنا مش بنطلع نواتج احنا بنطلع نسب اختلافات

- يعني هنطلع نسبة اختلاف ما بين الشقه الاولي والتانيه 0.85 فلما تيجي تضرب الرقم ده في سعر المتر الأساسي للشقه الي هو اصلا مش موجود معانا هيطلعلنا القيمه الصحيحه وبس كده .

=>هو ال Deep learning مقتصر بس على ال neural network ؟

- لا احنا ممكن نستخدم ال genatic algorithm من غير neural network خالص ب advanced algorithm وده يبقى deep learning عادي "neuro evaluation".

=>امتى استخدم devising clustering ؟

- لو انا معايا مثلا عميل عاوز أفصله حاجه على مزاجه فمعايا شوية packages نقي منها الي انت عاوزه فهنلاقي إن كل user ليه customization مختلف عن التاني , بس هو مختارها من مجموعة او أوبشنز موجوده عندي اصلا
- بس مش شرط كل user ي customise الحاجه بتاعته لا عادي ممكن ياخذ منالي موجود جاهز عندي وخلاص
- فدي حاجه اسمها personalization
- فاحنا بنستخدم ال devision في حالة إن احنا عاوزين ن personalize نظام معين
- وده بنعمله في حالة اضطراريه , يعني ال devising clustering هو توبيك advanced جدا وبيستخدم بس في ال personalised AI.

عن أبي هريرة رضي الله عنه قال: سمعت رسول الله صلى الله عليه وسلم يقول: "ألا إن الدنيا ملعونة ملعون ما فيها، إلا ذكر الله وما والاه، وعالم أو متعلم"