Week 3 Transcript

so uh this is material for week three um this is the learning outcomes i modified

0:09

it a little bit um because

0:15

um after I kind of prepared the material I saw that some parts were missing

0:22

so we will we have to kind of um you know just um organize

0:29

that um so the the the flow is this i

0:35

will talk about unsupervised learning in week three uh I will show some models such as K

0:41

nearest neighbors PCA uh um sorry K means not K nearest neighbors K means um

0:50

PCA assoc association rules um and subset selection

0:58

um we will talk a little bit more about like how to evaluate these unsupervised

1:04

models and then uh we will go very in a very shallow way uh we will talk about

1:10

recommendation systems or recommener systems okay so here is a

1:17

um um a figure that we should um recap first of all

1:26

uh we are this is a like the simple road map between supervised and

1:33

um that comprises supervised and unsupervised learning again we have more we have other types of learning uh but

1:42

but these are the ones that we talked about um um in week two we talked about

1:48

supervised and now in week three unsupervised so um if you have a labeled data in your

1:56

training data set then you can do supervised learning okay

2:03

um and and um if you think about building a

2:10

model a model of a prediction model let me let me be more specific a prediction

2:16

model there's this modify here that would continue into other stuff that we

2:21

won't see um but this is used for control uh but if you if you want to

2:28

build a prediction model you have to assess if your data if your labels are

2:34

discrete or continuous if it's discrete you're doing a classification if it's continuous you're

2:42

doing a regression okay and we saw this um in week two now

2:49

we're going we're going the other way we're going we're going to use

2:55

data to where where we don't have labels

3:02

right um so if you don't have labels if you if you cannot tell

3:11

um characteristics of the data apart a prior right

3:16

so you you don't have labels then you are doing unsupervised learning okay and

3:23

there are two tasks I would say two main tasks on supervised unsupervised learning um that

3:31

depends on how you are depends on the characteristics of the data

3:38

characteristics of the data if you think that your data can be

3:45

grouped into discrete groups then you're doing what we call

3:51

cluster or clustering right so so we're kind of um we are we are

3:59

clustering different uh data points because we we think or we

4:08

aim to of uh we um find similar

4:16

characteristics of these data points inside those groups okay now if you are if your data is just

4:24

a you don't know if there's groups like or this is not your intention

4:31

um your your you have lots of data points um that you don't want to

4:38

clusterize you might want to clusterize you might discover that you you can clusterize you have this common

4:45

characteristics in different groups but if is if you if you want just to

4:53

um if you want just to kind of learn and have insights on the structure of the

5:00

data not necessarily dividing the data into discrete groups um then what you're

5:07

doing is and and here is a nomclature uh difference um it's kind of not there's a

5:14

debate you what you're doing is dimensionality

5:19

reduction this is the most common i copied this image from from a

5:24

guy that uses the word embedding um

5:32

uh and feature selection so so you can you can you can hear all this all these

5:38

terms of uh tasks that you can do with unsupervised learning so there are two

5:44

main tasks that you can do with unsupervised learning you can group your data into group discrete groups or you

5:52

can the the word I prefer is you can do feature selection and dimensionality

5:58

reduction it's kind of the same task um that you can do with the unsupervised

6:05

learning okay so let me give you examples of

6:12

unsupervised learning unsupervised learning uh is very used um

6:20

in marketing uh uh for marketing strategies within

6:27

marketing um sectors of businesses mainly by segregating

6:35

customers and finding similar characteristics for customers and then

6:40

shaping the policies or the uh the sales strategy strategies um to each group to

6:47

each cluster of customers um and supervised learning is also uh uh

6:56

used like to discover topic discover or or to uh discover important traits of

7:04

the data so for example you might be looking at

7:12

um you might be looking at a data a data set where you have uh uh the there in

7:22

this data set they log the delays of airplanes right and you will you will

7:29

notice that delays are

7:34

um uh there's a column on so this is your data set there's a column on delays

7:41

the the number in minutes of of the delay of each aircraft of each flight

7:47

and there is a other uh so this is in minutes and there's another

7:54

uh feature another attribute that is whether you had the delays or not so

8:00

delays yes or not or or no so you would imagine that if if here's if if there's

8:08

a number here like 10 minutes delay this should be yes right but it's very common

8:15

that you actually you have a missing value here or you have a no right and

8:21

then you will discover by kind of assessing this data

8:28

that uh you you're going to discover that the

8:33

data is um this is my dog uh that the data is

8:40

uh actually delays are defined find more than 10 minutes of delay like

8:46

like the you you are only you only report of a delay after 10 minutes so

8:53

you you might have um you might have data and you can you can actually track

9:00

9 minutes for example but it's still not marked as delay and you and and you um

9:06

and you can actually learn this kinds of u you have this insight discovery of of

9:14

the data by going through unsupervised learning usually we use uh unsupervised

9:22

learning when we don't know what we're looking for uh we want to we can group

9:27

the data this is more of a def more of a task that is defined but if you want

9:33

just to have insights and discover and you know uh have this this uh insights

9:41

on what the data means we use unsupervised learning that is why

9:46

unsupervised learning is the method of learning um that is very used into the

9:52

uh feature um feature selection extraction and

9:57

engineering uh feature selection feature extraction and

10:05

feature engineering step of the data science cycle right so if you remember

10:11

you have the data you have a pre-processing step then you have all this feature uh this discoveries these

10:19

insights on the data then you will take these attributes to your models and do

10:24

whatever task you want to do okay so it is true that you will actually use some

10:31

unsupervised models models here right to do this

10:37

kinds of discovery within the data okay and these models here are more for a

10:44

kind of supervised um task okay we we we see unsupervised learning

10:51

in genetics research if you want to group uh genes that has same expressions

10:57

uh image segmentation so you want to group pixels of a picture that are part

11:02

of the same object uh again market research mag medical imaging as grouping

11:10

cancerous images um uh parts of the image and even social network analysis

11:18

uh and like what's the characteristics of people that does search certain action in social uh networks okay so

11:27

we're going to try to distill this a little bit more and we're gonna we're gonna start

11:34

talking about clusterization and it can be mis

11:41

um misinterpreted as classification but it's not so classification if you can see here we

11:48

have two different classes right so cats and dogs and what we want is we want a

11:55

um a decision boundary want a model that can separate them as good as possible

12:02

right so when new data comes in you can actually classify the new data but you

12:08

have the labels now here uh in clustering you you don't have the labels

12:15

meaning that you just for example you would have just photos of cats or dogs

12:21

right and um so all the points are blue I don't

12:26

know what they are and obviously that this example we know cats and dogs but

12:32

imagine that you're seeing two kinds of reptiles that you never saw before and

12:38

you don't know the names of them but you can see that one of the reptiles has

12:44

more of a greenish like the a greenish um um skin right and

12:54

um maybe a uh a kind of a mane and the other does have like a yellow yellowish

13:02

uh u um skin with no mane right so if this is

13:08

the skin color near to one is green near to zero

13:13

is kind of yellow and here is the zero is no main one is main so you can

13:21

see that there are there are several pictures that

13:26

that this this data points are are more similar to each other right than this so

13:34

you can h kind of make groups okay obviously there are uncertainties but

13:40

this is the goal of clustering okay

13:46

um and sometimes it's very like very separated and easy to see like this

13:53

example that I gave but most uh I would say most of real pro real life problems

13:59

the the data is super messy so let's say that here you have three different uh

14:06

reptiles and the data is super messy and

14:11

and uh it might be that it's not visually it's we cannot visually

14:17

differentiate the uh the animals is super hard for humans to do but if we

14:23

have a computer algorithm that can perceive that this points here are similar to each other this uh dark blue

14:31

is similar and that uh uh light blue is similar you you can actually uh you will

14:40

have the insight that you have three different reptiles in um in the data so

14:47

let's talk about the most common uh algorithm for clustering the name is

14:54

K means and again as just the K nearest neighbors in classification this K is a

15:00

hyperparameter is a parameter of the model like you will have to choose uh so

15:07

it is necessary to have at least a hint of how many groups of people have

15:16

in you have in your um in your data right so a priorize you

15:23

have to kind of have a notion of uh how many reptiles you think there are in

15:29

that um in that data set uh so let's say that we have this uh

15:37

this data set here so I'm plotting the points and the

15:43

the K means algorithm uh starts like this you assign a c a

15:51

random centrid to the number of clusters of choice so we are choosing a three

15:59

uh three k equal to three meaning that I think there are

16:05

three three um three different clusters and I randomly assign the

16:12

centers of this cluster so I assigned a green cluster the center of the green

16:17

cluster here center of the um red cluster here center of the blue cluster

16:24

here okay and the algorithm is iterative so after you choose the

16:32

centrids the algorithm calculates the distance of each point to each one of the

16:40

centroidids and assign the point to the group of the closest

16:47

centrid so what I mean by this is for example for this point here I measured

16:54

the distance to the blue to the red and to the green centrid and obviously that

17:00

this point is closest to the blue so I labeled this as oh centrid blue okay so

17:08

it it it pertains it consists of the centrid part of the centrid blue uh part

17:15

of this group blue okay um same thing as this here for

17:21

example if you measure the the closest distance is for the

17:27

green and I assign it green okay so after you you chose a centrid position

17:35

and assigned the points to each of the groups you do the you um you move the

17:41

last step is to move the centrid to the average or to the middle point of the

17:50

group so what I'm going to uh what I did is I moved this this here i moved to the

17:58

center of the group right so uh now it's

18:04

here same thing to the red the red I got the red and I moved to the center of the

18:10

red okay um and it's here now and you

18:15

can see that because of these two little um points the center is not here but

18:21

it's kind of this two points pushes to the left a little bit and finally the

18:27

green as well okay so I I moved uh this centrids and that's one one step of the

18:35

algorithm and now you iteratively do this all the time so you have new

18:42

position of the centrids again you do step number two that is assign the

18:48

points to the closest centrid so now you can see that both of these points that

18:54

that were red now they're closer to the blue point

19:00

right and then um so you have a better

19:05

uh a better assignment and then you finalize with step three that is you

19:11

move the centrids again to the middle of the distribution of points so you have

19:16

blue in the center green in the center and red in the center and you can do

19:21

this lots of steps and in the end uh hopefully you will have three different

19:28

groups um that is what visually we would guess right we would guess this as a a cluster

19:35

meaning that they have similar characteristics let's say that this is feature X1 this is feature X2 i don't

19:42

know color of the hair color of the um or uh if it the height of the person so

19:49

this people here have same color of hair it varies a little bit but not that much

19:56

and uh they have like a similar stature as well this ones are another group this

20:02

one's another group but again you have to have a good guess in the beginning of

20:08

u the clustering algorithm if you were to choose K equal to two it might be

20:13

that in the end after um you would assign one centrid here randomly one

20:19

centrid here and in the end doing this you might like end up having um one

20:28

cluster here and this two clusters there okay

20:33

so the the higher the K the more granular you can go in terms of

20:39

separating into groups but if you have a too high K

20:46

uh then you might be very like you might be um in in in the limit you might be just

20:54

thinking about each instance um individually

21:00

okay there are uh other algorithms that we're not going to discuss here uh that

21:08

that can actually uh find the number of K's automatically

21:15

they it guesses the number of K this is the DB scan uh if you are interested we

21:22

can leave resources for the students to kind of read about how the DB scan algorithm like works it's a

21:30

modification of this iterative process and uh it's super useful because

21:37

DB scan can actually find uh different formats of clusters that

21:44

sometimes K means can't so K means

21:50

um because it's in a in a rectangular coordinate you will never be able to if

21:56

you think about this example you'll never be able to even if you assign two groups K equal to two um K means we'll

22:05

never be able to separate the internal circle as one group and the other one as another

22:12

um and and um because the centroidids of both groups have the same center it it's

22:20

it's not possible right now db scan this modification uh it it can do several several crazy uh

22:30

better uh clusters assignment and it does it automatically so it did

22:36

automatically k equal to two here the number of clusters equal to two here uh

22:41

as well it made uh equal to two here it make equal to three

22:48

and here uh here this algorithm just said no uh this is just we don't have

22:53

groups in this data so DVC scan is very powerful but K means is very it's

22:59

computationally very effic efficient and also uh in in data sets where you kind of

23:07

have a hint it does a pretty good job okay it does a a pretty good

23:15

job okay now so that's clustering and it's simple

23:20

as it is it's just grouping uh people with the same uh with the same

23:27

characteristics so for example let me go back just a little bit in a banking example where you are looking at the the

23:35

profile of online users for a for a bank app right you will have

23:44

several you will have a database with several characteristics age uh like if if it's

23:53

um um if the person is taxsavvy and and you

23:58

can you can have this information by doing uh quick surveys in the app uh you

24:06

can um uh if it's it's if the person is female or male um and so you have all

24:13

these characteristics and obviously that we cannot plot

24:19

uh like all of this like age is here tech savvy is here we cannot plot all

24:26

the features because it's more than three-dimensional but it might

24:31

be that and I'm just plotting two of them um

24:37

that that you have two do you have groups that are very very

24:44

distinguishable and for example you can have a a group

24:52

that they are there are young they're tax saving tax and they're

24:59

female and you can have other group of people

25:04

and uh you can see that this this

25:10

profiles they can help other algorithms of for example predicting

25:16

uh investment actions right uh if they're going to invest their money or

25:22

not okay all right so let's talk about dimensions

25:28

uh when we talk about dimensions we're talking about the number of features just as I said age tax if it's the

25:36

person is tax savvy it's female or or male uh when did they open the count so

25:42

you have features are the columns or data set and we call this dimension so

25:48

if you have a six if you have a a dimensionality of six you will have to

25:53

have six columns six features in your data set and the rows are the instances

25:59

or the observations okay if you want to predict the price of a of a

26:06

um the price of a house uh for example you can you can take into consideration

26:12

the square footage the number of bedrooms the location the age of the house the distance to the to downtown

26:18

and the number of bathrooms of the house so you're in in a sixthdimensional

26:23

um in a sixthdimensional problem now the in in in in uh in the in the

26:33

data science cycle you store the data you come up with the

26:38

data then we call there's a step called pre-processing and um there's another

26:47

step called like the feature engineering and uh feature kind of kind of feature

26:54

related tasks or feature tasks this this both of this sometimes

27:02

you you see them separated but both of this can also be sometimes

27:08

um they are part of a block or a step named

27:13

pre-processing so uh people put the feature task inside the pre-processing task meaning that

27:20

this is before before feeding your model okay so this this uh dimensionality is

27:28

very important in selecting the features in se selecting the features or coming

27:34

up with new features or defining which features are most important it

27:40

might be that the distance to downtown is not that effective in describing the

27:47

the the price of a house as for example the age of the house so it might be that

27:52

you're just going to drop the distance to downtown and you will just feed your

27:58

model for example your prediction model um your your supervised model prediction

28:06

model u with the most important features

28:11

okay so we saw already that feature engineering is kind of creating new

28:16

features that are important uh and we will talk about feature scaling and and

28:21

dimensionality reduction um a little bit more so we're going to go a little bit more into the

28:28

pre-processing world um and we will see that there are there

28:33

there are uh unsupervised methods that we can that we can use um to do these

28:40

things okay so I'm going to start talking about where is

28:46

it about uh data types okay so when when you see

28:55

your data set your database um you have for example this it's uh

29:03

it's the the employer it's the the employee data set you have the age of

29:10

that person uh which company the employer right and

29:17

the salary okay so you can see that different data sets have different types

29:24

of of um of columns so in

29:31

this in this column here we have a numerical

29:36

value that is continuous right so it it means that it

29:43

can assume whatever value you think of

29:48

age is also a numerical uh value but it is discrete meaning that

29:56

you you cannot have any value that you can imagine you cannot have um

30:04

44.99 years right um you have either 44 or you have 45 okay so you jump from

30:14

from number to number you have the employers this is a what we call a

30:20

categorical data type so every number or this is actually

30:27

a a word right employer one employer two employer three it it it could be like I

30:33

don't know um um

30:40

um uh whatever like it could be General

30:46

Motors Ford employers from this this produ car

30:51

production companies um Dodge and and so

30:57

on and these are words there are cate there are different cate they're

31:03

categories right so we call it categorical so this is just one example

31:09

but the in the end of the day we have I'm going to list out the

31:15

the types of numbers that we have nominal um or what we call categorical so

31:22

categorical or nominal categorical for example hair color

31:29

well hair color can be blue uh sorry can

31:34

be um hair hair color can be blue yellow

31:39

brown red uh uh orange purple if the person is

31:46

kind of um dying uh her his hair but you will have

31:54

a finite amount of values of different

32:00

categories okay you can have a hundred hair colors but you still have you you

32:06

can still count it's they are discrete categories a finite number of

32:13

categories occupation is also categorical if you're an engineer if you're a business um a CEO if you are a

32:22

um electrician and so on there is a type of

32:30

um variable that's called binary it's actually

32:35

a it's a um it can also be used to point out to

32:43

categorical so zero is sick not sick and one is sick but the the number per se

32:50

that will appear in the data set is a number so it's a numeric um it's a

32:57

numeric value so this is a

33:03

numeric but it only assumes with only two values with only

33:08

two possibilities right and we will see that this is actually a

33:14

subcase of the discrete numeric values we have ordinal

33:22

um data uh ordinal data types right so for example

33:28

uh ordinal they are categorical um data but where the order matters

33:34

right so let's say that you're uh you're analyzing drink sizes you have P ML

33:43

um sorry SML small medium large but the

33:48

the the categories like they have a a

33:54

um they have they are different names right such as hair color blue uh uh blue

34:03

brown yellow but here the names are overloaded with a order meaning that

34:12

the small is less than the medium and large is um larger than the med medium

34:19

as well as customer satisfaction right so it's um it's you can have a a

34:27

category of I'm super happy not that happy A B C

34:33

D um and but there's there's an order meaning that A is more valuable in terms

34:42

of B of B and C and D okay so that's the ordinal kinds of uh data and they they

34:49

are they can be categorical they are categorical okay um you can also think about

34:57

customer satisfaction from 0 to five right uh and but that that's numeric and

35:04

and numbers already have um embedded into them the concept of

35:10

order right 0 is less than five but um when we are talking about ordinal we

35:17

have the the like we have we we need to know that this is this is our these are

35:23

categoricals they're discrete um like labels right and but they they

35:31

are overloaded with the sequence and the numeric

35:38

uh you can have discrete so this is inside here discrete

35:43

and continuous discrete is for example age right you have a finite number of of

35:54

um not finite uh you have a sorry let me pause this again you have a a discrete

36:01

you don't you don't assume any number that you want you you have like jumps um

36:08

in numeric uh in the numbers

36:13

um and continuous then then it could be anything right for example the price of

36:20

a house could be any kind of number that you think of okay all right so given all this um in

36:29

these things right when we when we when we

36:36

um when we get a data set we have to inside the pre-processing

36:43

steps we have to analyze the data types and deal

36:48

with problems that can arise one of the problems is missing values right so

36:55

missing values are a problem and the missing values can come can come because

37:03

you had some error in acquisition or um the person did not want to answer the

37:10

question or there were some errors in a sensor i mean there's several uh reasons

37:17

why you don't have missing values okay and missing values

37:24

uh can be tricky okay they they actually can get in the way of applying your

37:30

model because your your model needs that val the value coming in as an

37:36

input but it also can bias your analysis and

37:42

um and it can and can make your model perform very poorly

37:48

so we're going to deal with the the challenge of missing values and the

37:53

challenging of normalization okay so what is that in normalization is dealing normalization

38:00

is something that we do when one of the features is much higher than the other

38:05

one so for example here the salary uh you can see that

38:12

$73,000 it's it's an it this is a numeric continuous column this is a

38:19

numeric discrete for the age and but but the numeric value of

38:26

salary is so much higher than the age and in the these preprocessing analysis

38:34

it might be that your algorithm kind of disregard the age because this is so

38:40

this is smaller to too much smaller than the salary and that's not what we want

38:46

we want that the age to participate of the prediction of the of the salary or

38:52

to to the the prediction of whatever we're we're predicting okay so we we

38:59

will do then the normalization so two concepts here regarding two different uh

39:05

problems the missing values of a data set and the uh ND

39:13

normalization okay so let's start talking about missing to modify something uh actually

39:22

uh this is I changed from normalization to scaling because normalization is one

39:27

type of scaling okay so we have to uh scaling is when we have this difference

39:33

numerical differences uh between the attributes okay so let's

39:40

talk about missing values first missing values they can be unknown they can be

39:50

unrecorded they might be irrelevant right so um if you what one thing that you can

39:59

do with missing values if you have a whole column of missing values meaning

40:04

that if you have all of the instances or almost all of the inst instances with

40:11

missing values then it means that this particular column here is not it's not

40:20

informative and you can just remove the instances where you have missing values

40:25

right so um you have missing values missing values missing values missing if

40:32

if if this is a homogeneous distribution where most of it is missing you can just

40:38

remove this column because this column is not relevant now

40:47

um so different um but this is one type of assumption if you have a if you have

40:54

a a uh a a data set when you have just a few missing values and you can remove

41:03

those instances right let's say that you have enough uh

41:08

instances you have a pretty large data set and you can remove then you just

41:13

remove them they won't they won't do uh they won't they won't make a difference

41:19

to this distribution of the column okay to the statistical distribution of the column

41:27

but anywhere in between these two situations it's tricky why well because

41:32

you can if you omit the attribute you might lose important

41:38

information um if you take out the the instances you might

41:46

lose lots of data so omitting the the attribute and removing

41:53

the instances is just when the column is all

42:00

missing like the whole instances most of the instances are

42:05

missing or just a few were missing okay anywhere in between you can uh you can't

42:13

guarantee so one way to deal there are several

42:18

ways to deal with missing values uh I I'm leaving here two very good resources

42:26

one of this one of the strategies is actually for example for

42:32

numerical columns uh let's say um let's

42:38

say that this the square footage of a house right you have you have some

42:45

missing values and you cannot remove these instances because you don't have

42:50

enough what you can do is you can take the like nine 90 100 200 then you have a

42:59

missing value a missing value and then 300 right one thing that you can do is

43:07

try to with a machine learning model regress this missing numbers here as

43:13

this was this is unseen values uh from the other attributes right so

43:20

let's say that here you have the number of bathrooms and the age of the house so

43:27

you can try to create a model that learns from the data you have

43:34

and it's reg it it it it it finds a relationship a

43:40

regression between the square footage and the age and the number of bathrooms

43:49

and then you come here and then you just complete this this is one of the ways you can uh deal with missing values the

43:57

other way um if if you're miss if you have

44:02

categorical data is clusterizing data that is similar and then so let's say

44:09

now that um um it could be categorical or disc

44:17

numeric discrete so let's say that now you're missing number of bathrooms you

44:22

have four you have five you have eight you have nine so one thing that you can do is

44:28

clusterize the the data the all the instances that are similar and then if

44:34

this is similar to uh this one then you assign it a five

44:43

right so you can you can use clusterization to tackle missing values

44:50

okay and again for for clusterization it might be too obvious that I'm saying

44:55

this now like oh this is similar to that uh like number of bathrooms and age it's

45:01

it's easy to see who is similar to who but in big data sets where you have lots of dimensions that is not simil that is

45:09

not that easy anymore and that's where um clusterization algorithms will help

45:16

you a lot but so this is like what we call

45:21

this is what we call imputation imputation is we're impu imputing we're

45:27

we're doing the imputation of the missing values and we have all this we have we can use machine learning to

45:33

impute this values uh either supervised or unsupervised models

45:40

um you can also do imputations with the mean right say so say that I don't know

45:47

this the the square footage uh but I'll just assume the mean of this

45:53

this of all the square footages it depends on what you're doing and it

46:00

really depends on uh a little bit of experience of the data scientist the data analyst that um to to come up with

46:08

this the better imput computation method um but the the overall goal is not to

46:16

modify the the statistical distribution of the columns is to kind of maintain

46:21

the statistical distribution untouched um the other thing is if this missing

46:29

value um it might be that it's not irrelevant

46:34

and it's important for us to know that some of the instances has missing values

46:40

because of some reason okay and what what we also do sometimes

46:47

is just we engineer so we create a new feature called is

46:54

missing and we will we will uh like for example let's

47:01

say that the square footage is missing and we will flag one if it's missing

47:07

that information on the square footage or not because it might be important it

47:12

might be that the Rialter uh uh kind of ignored square footage for

47:21

uh houses that are that are near downtown and you're you're not aware of

47:28

the this but you will discover this if if you don't ignore the missing values

47:35

if you assume that the missing values is not irrelevant and you kind of create a

47:41

flag for it okay so there are several methods to deal with missing values and

47:48

we we we will you can take advantage of machine learning or statistics

47:54

um either supervised unsupervised or classical statistic statistical methods

47:59

to to do imputation or you just ignore if you ca if you want or if you if you

48:06

don't think it's if you think it's not um irrelevant you you can engineer a

48:12

feature that shows um why a person and then you can train a

48:18

machine learning model to discover when a person will will miss this information

48:24

and then you will discover for example that it's because it's near downtown so this is the part this is

48:31

part of the data mining process the pre-processing um feature selection extraction

48:38

engineering part of the data science cycle okay it's very interesting to see

48:43

both of these videos it's very important so going back now we're going

48:48

to talk about scaling and um um feature

48:56

scaling uh there are two types of feature scaling normalization or

49:01

standardization and it's the process of transforming numeric features to similar

49:07

scale um typic typically to prevent features

49:14

uh with larger like values from dominating the learning process right so

49:21

uh for example here as we already discussed the numbers for salary are much larger than those for

49:28

age and it could just consider age while fitting just the salary for whatever

49:35

target we're trying to predict and uh scaling is just normal uh so one of the

49:43

scaling is is is um is known as normalization it means that we will

49:52

transform the age um that was numeric discrete to values

49:59

between zero and one so if if 48 is your max value it will assume one if 44 was

50:07

uh 27 is your minimum value it would assume zero and you will distribute your your

50:14

data in between that same to the salary so having this done for example this nor

50:22

uh normalization um will make your model

50:30

um kind of take the features in the same way you're not biasing a feature

50:38

um or or whatever or uh like more than

50:43

another the other type of of scaling is called

50:49

standardization it means so if I were to create an image here so this is

50:54

standardization uh for example the salary standardized

51:02

if you the you will come up with numbers that if you plot these numbers

51:11

in a histogram you will find

51:16

oops you will find that the mean of these numbers are

51:21

zero and the standard deviation is one okay so we we we try to we there's a

51:31

transformation of this column into a col into a normal variable a normally

51:38

distributed variable and I'm not going to discuss here this the methods that we use to do this because most um most

51:47

softwares data mining um machine learning softwares do this automatically

51:53

but I can also leave a resource for for you to know how to standardize and

52:00

normalize okay so after we've done this pre-processing

52:09

um here what we kind of miss we we we kind of addressed missed values we

52:14

scaled properly the columns uh sometimes we we have to to do some feature

52:20

engineering for the miss the missing value so kind of we get into this this little box here already but after you've

52:28

done this you you we will talk about now something that you can do that is

52:35

dimensionality reduction so what is dimensionality

52:40

reduction um here uh in this text here you in the

52:47

the right in this grid of images with the text here you can see that you have two images you have an um you have an

52:55

Airbus uh A380 on the on the bottom

53:03

A33 and you let's say that you want a classifier that classifies uh one and

53:09

the other you don't need to feed the classifier the pred the the the

53:15

classification prediction predictor with all this pixels you can see that you can

53:22

you can feed less pixels meaning that you can

53:29

um if you you have a data set you can feed less columns right uh and

53:37

your your algorithm will still be able to um to classify and it will be

53:45

computationally more efficient okay so

53:52

uh there is one reason to do dimensionality reduction is exactly for

53:58

computational purposes there's another reason to do dimensionality

54:03

reduction that is sometimes in in highdimensional spaces

54:10

if you have if you have a highdimensional problem meaning that you have lots and lots of feature lots and

54:16

lots of attributes and columns of your data set machine learning

54:23

uh struggles to find patterns right so

54:29

the the p when you grow in dimensionality the

54:35

patterns they kind of get more scarse right they're not so clearer to

54:41

see and um and it kind of turns out to be a

54:48

problem finding a a um a needle in a hay stack right so the

54:55

the models kind of struggle this is called dimensionality the curse of

55:02

dimensionality so the curse of dimensionality and uh this is one thing

55:09

that as data analysts or scientists uh we need to be careful because

55:19

uh and you building a prototypes and the students business students but building

55:26

prototypes they have to know that if they if they have too uh few attrib

55:33

attributes they can they might not be informative for the model like if I keep

55:38

reducing the images at some point the model won't have enough information but if I give lots and lots

55:46

and lots and lots and lots of information uh it might be that the model also

55:52

struggles because the um the patterns are not uh are not that

55:58

visible okay Um so I will leave a so I will put a

56:05

resource here because this is good i mean if they want if it's it's out of cur

56:11

curiosity so usually when we're we're talking about uh big data it's it's

56:18

intrinsic and natural that we have lots and lots of data okay and uh we have lots and lots of

56:27

columns or features so one of the things that we have to do before feeding our models or prediction

56:35

models for example is to do dimensionality reduction and and I will

56:41

talk about uh one algorithm here named PCA uh PCA stands for principal

56:50

component analysis so principle

56:57

component analysis okay so just uh before moving to PCA uh I I I've inserted the the

57:06

resources now for the curse of dimensionality and I I want

57:12

to in a very shallow way uh present what that means so I found a a very good

57:20

resource here um and I'll I'll try to go over the

57:25

resource uh but we can leave this as um an extra video uh I I just want to

57:34

leave the explanation here because if we want to um uh to generate our own

57:40

explanation I think we could use uh this kind of reasoning to

57:46

explain so here we show um let me go back a little

57:52

bit so what we're doing here is we are showing and this is a code uh where the

58:02

guy has a has data and he will show

58:11

um this data in just one second let me find

58:18

it all right so here here's the start so let's say that you're using a k nearest

58:24

neighbors to um to classify

58:30

uh uh class uh to classify two different uh classes

58:37

right blue and red in a one-dimensional problem so let's say that you're only

58:43

thinking about the x-axis so you could you could draw a decision boundary here

58:51

and obviously that your model would get the like this little dot it would say

58:57

that it's blue because it's to the right and it's actually red but you will do fairly well so it is

59:05

obvious that you have two separable classes in

59:12

one dimension and if we're using k nearest neighbors uh we are actually

59:18

assigning so let's say where the mouse is here if a new point comes in you will

59:26

assign the blue class to to the nearest points right so if you you would assign

59:34

this new unseen point as blue okay now if I so what what this guy is

59:42

doing is he's plotting the distance between all these points the histogram

59:48

so how how does this the distance uh between this points um

59:56

um is distributed so you can see here that there are several points that if

1:00:03

you measure the distance between the points there are several points that have a small distance between them

1:00:11

meaning that they are neighbors there's lots of neighborhoods here um and that's

1:00:18

why K nearest neighbors works and um um it would be fairly good in terms of

1:00:25

effectiveness in in separating the uh these classes again uh this little point

1:00:31

would be classified as blue because the the neighbors are blue you would you would getting be getting it

1:00:38

wrong but I would say you would have like 99% accuracy here um so when we

1:00:45

plot the distances you can see that for example this point has a long distance

1:00:51

to other points for example to this other red uh but still have a short

1:00:57

distance to this one so in the end of the day most of the points have neighbors so have small distances and

1:01:05

and just just um some of the points not many have long distances okay so that's

1:01:13

this is what's depicted here so you have most of the distances

1:01:19

um are near zero or are are very small and as you go uh like as you grow the

1:01:28

distance so here in the x-axis is the distance the greater the distance the

1:01:33

less amount of points um have big bigger distance between points right now what

1:01:41

happens when we analyze uh this this data introducing a new

1:01:49

dimension so let's say that we have again two

1:01:54

classes the blue and the red but now in two dimensions so I kind of added a

1:02:00

dimension to the problem and you can see

1:02:05

that what what's going to happen to that distance graph is that we still h we

1:02:11

will still have vicinities or u neighborhoods points lots of points that

1:02:17

have um small distances and just few points that have

1:02:25

longer distances uh but you can see that the data is more

1:02:32

spread out it's not as close each point is not as close to each other as one

1:02:37

dimension so if we plot this

1:02:42

um what will happen is we'll still have short distances

1:02:49

um but this like the if you compare to to the graph

1:02:56

before this uh here you will see that it's it's less

1:03:05

uh there are less packed right so this peak here got like

1:03:12

near near uh zero in the very small distances got down right so what's

1:03:20

happening is that as you include a dimension you are actually making your

1:03:26

data sparse right so so that pack that pack of data points that for example

1:03:35

KN&N would excel uh it's still excelling here in two

1:03:42

dimensions but you were imagine you would imagine that as you add more

1:03:47

dimensions this data points starts be start being uh even more spread apart

1:03:54

right so at some point you will lose this kind of pack characteristics and

1:04:00

your can nearest neighbors won't even know uh there won't be like

1:04:05

neighborhoods to to um assign the the new points new points

1:04:13

coming in so we don't we we won't plot obviously in uh we could plot in three

1:04:21

dimensions but not for four 5 10 because we cannot visualize it so what the guy

1:04:27

did is he started plotting the distances so for

1:04:34

example for uh for three dimensions

1:04:41

uh for three dimensions you can see that uh let me see if I can get the image

1:04:47

here so for three oh man okay so for

1:04:53

three dimensions the you can see that the smaller distances like near zero got way

1:05:02

uh less and now the points are getting more spread obviously still we have uh

1:05:11

also a small amount of points that have great distances but but you can see that

1:05:17

we are unpacking the data and he keeps plotting so dimension four dimension

1:05:24

five for example dimension 10 you will see that we have a very low amount of

1:05:32

points that are close together and we have some points that has distance

1:05:38

between two and three and here between five and six um most of the points have

1:05:44

a a a fair amount of distance between them um but we lost that that pack

1:05:53

characteristics in the end of the day if you bring it to the limit to for example

1:05:58

a um 200 300 500 dimensions

1:06:05

uh you will have points that distance like for example

1:06:11

between 18 and 20 or between 40 and 45 42

1:06:17

uh but they are distant apart I mean they they have these distances but you can you can imagine that these points

1:06:24

are very spread spread apart they lose their their pack characteristics

1:06:30

okay and you can think about also uh let's say that yours you want to

1:06:37

retain this is just a simulation for us to understand you want to retain 10 samples

1:06:44

uh for uh a unique combination of variables so let's say or attributes so let's say

1:06:50

you have one binary um variable and I'm not here's better

1:06:58

you have one binary v variable so you have uh like one and zero and zero and

1:07:05

one so you have two different combinations of this column so let's say

1:07:10

that you want to retain 20 samples okay and and this is arbitrary it could be

1:07:16

whatever I just want to show you the trend if you if you

1:07:22

now if you now add one variable so you you first had one column now you have

1:07:28

another column and so you have four different combinations you can have uh one

1:07:38

zero zero zero one 0 0 1

1:07:45

uh 1 0 1 0 and 0 1 0 1 okay so you have

1:07:50

four unique combinations so you if if you have if

1:07:56

you want to retain 10 samples per unique combination you would have 40 samples needed and if you keep going right so

1:08:05

what what we're what we're showing here is that you added just one variable

1:08:12

and you doubled the number of samples that you want to retain so if you put

1:08:20

that into perspective if you if you kind of keep

1:08:25

increasing the number of variables the number of samples needed and this is in millions so this this is

1:08:32

why it seems like constant here but it's growing a lot but when you get near 17

1:08:39

you explode this is an exponential growth so this is the why we call the uh

1:08:46

the curse of dimensionality right but the other resource is also very good and

1:08:53

it shows like some 3D plots to kind of understand that okay so now let's talk

1:09:00

about PCA um so most of the time even 10 10

1:09:08

features is um for some models is confusing

1:09:16

uh deep learning as we discussed in the week in the past week kind of gives

1:09:21

us a a solution or is one of the most robust models that accept lots of

1:09:29

features but even um even even that uh could could be a problem for deep

1:09:37

learning uh but for most of the models having too much dimensions is a problem

1:09:43

so we will work in dimensionality reduction before

1:09:50

feeding our models so we one of the algorithms is called PCA and it stands

1:09:56

for principal component

1:10:04

analysis and uh uh PCA is a very intelligent

1:10:10

algorithm and it's also unsupervised we call it unsupervised it will learn from

1:10:15

the data from the characteristics of the data um and it will in the end of the

1:10:21

day engineer a new feature okay that is

1:10:29

uh more informative and it will disregard

1:10:35

uh previous previous features so that we have a dimensionality reduction okay so

1:10:44

for example I will give you an example where where two variables turns out to

1:10:49

be or two features turns out to be one so you dropped a dimension and this one

1:10:55

is more informative or is is is um it carries the information that you need

1:11:02

okay so let's say that we're trying to predict types of fish based on several

1:11:07

features like length height color number of teeth or something like that

1:11:14

but you so when you're looking at the correlations of the different columns of

1:11:20

your uh data set uh we we find that height and

1:11:28

length are strongly correlated right and

1:11:33

um including both what like because they're so correlated meaning that they

1:11:40

uh um they have like the same behavior uh including them both uh might not only

1:11:48

be not helpful but it could also hurt uh the model in terms of

1:11:56

dimensionality so what we can do and what PCA does it uh feature it engineers

1:12:03

a new feature that is a combination of these two correlated

1:12:09

um features okay uh and it's it's very common to use

1:12:15

this in large data sets and and obviously will allow us to dramatically

1:12:21

reduce the um dimensionality so the way we do that is

1:12:30

um the PCA will al actually engineer a new feature called like the the shape

1:12:38

right the a shape feature and

1:12:44

um the shape feature will kind of be a combination of length and height because

1:12:51

they are actually kind of conveying this length and height because they're so

1:12:59

correlated they are kind of conveying the same information right if you increase length you're almost linearly

1:13:07

increasing height as you're seeing here so so they mean like they they grow or

1:13:15

or decrease together right this is what

1:13:20

correlation means okay so what PCA does to create this shape feature is find the

1:13:30

direction in which or or find uh a uh a

1:13:38

direction of where uh so find a direction in which

1:13:46

most variance in the data set is

1:13:52

uh is ex is retained so in this example most of the variance

1:14:00

um in in the fishies is in the diagonal okay so the

1:14:07

the fishes have the data is very spread if you

1:14:13

think about spreadness or variance this this data is very spread in this axis

1:14:21

here that we are calling that we are engineering calling first shape score

1:14:28

okay or the shape score and um this this is called the fir the

1:14:35

principal component and this will become our new engineered new engineered feature okay

1:14:43

and the second uh component the second principal component is orthogonal to the

1:14:52

first shape so you can see that um you can see that the the data is more

1:15:01

spread here in this axis right so this axis

1:15:08

carries the information the ver um explains the variability of the

1:15:13

data best than any any kind of feature and the orthogonal one meaning

1:15:22

the data here this orthogonal one

1:15:28

uh is the one is the direction where it it it only explains the smallest

1:15:35

smallest part of the variation right because it's orthogonal to the

1:15:41

first so the the first princip the first

1:15:48

component the engineered the new engineered value

1:15:54

feature new engineered feature or column

1:16:00

will be these values here in this in this line sitting in this line because

1:16:08

that's what explains the most spreadness of the data okay and the second

1:16:14

component is where it explains the smallest variation of the data you can

1:16:19

see that the v the the v the the data varies up and down right up and down i'm

1:16:26

not talking about this spreadness i'm talking about this spreadness so um it

1:16:34

it it varies in this orthogonal direction in this orthogonal direction

1:16:39

here but it varies so little okay so what we're going to do is that we're

1:16:45

going to remove can we will exclude this B12 B uh

1:16:52

sorry B2 axis we're going to exclude this feature in this data set and just

1:16:58

use the shape okay so now we're not using length or height

1:17:07

we're just using the the the

1:17:12

the principal component the first principal component right which is the

1:17:18

one that explains best the variability of the data um and this is

1:17:25

PCA okay i say this new engine fe this new

1:17:33

um so let's say that you we will generate one column here named

1:17:40

shape and we're going to delete both of the length and height

1:17:47

columns and this axis of shape it has its zero here and it grows 1 2 3 4 so

1:17:56

you're you're you're you will this is your new reference line

1:18:03

right so this is what will appear here this this points

1:18:08

uh will appear and it's it's this new number line that you're um assuming for

1:18:16

the shape right so you're not you're you're dropping you're dropping this columns

1:18:22

and you're generating a new column where your your new feature is actually this

1:18:28

this line here

1:18:33

okay uh I think that for the course it would be very good if we could um kind

1:18:39

of recap correlation at some point uh it could be an extra resource

1:18:47

uh it could be some kind of a reading on statistics

1:18:54

but it's it is it would be very important to uh talk about

1:18:59

correlation okay uh I won't I won't talk about correlation now because the the

1:19:05

lecture would be immense but I think that people will have to have this

1:19:10

concept on correlation um it's it's good to have that

1:19:18

okay so the other way so we saw that during the

1:19:25

pre-processing slashfeature uh step feature selection extraction

1:19:32

uh of of our cycle life cycle we can we

1:19:38

can do we can we will have some challenges first of all we will we will

1:19:43

have to deal with missing values we will have to deal with

1:19:52

dimensionality and we can do that dimensionality reduction using PCA for

1:19:58

example we can create new features doing feature engineering

1:20:04

um like we can manually generate a a feature if we know like the if we have

1:20:10

domain expertise but we can also do something

1:20:15

called a feature selection and that's what we're going to discuss now so what is feature selection okay so

1:20:25

feature selection is actually uh uh picking the best features and

1:20:32

eliminating the worst features right so it is a kind of dimensionality

1:20:38

reduction but it you uh we call feature selection

1:20:44

um PCA is a method of feature um it's it's not a fe it's it's a method of

1:20:51

dimensionality reduction but we don't say it's a a method of feature selection

1:20:56

because you're actually engineering a new feature from the principal component

1:21:02

um feature selection you don't engineer a new one you just pick the one the ones

1:21:07

that you want from the existing data one of the ways to do that um it's

1:21:15

it's uh it's called correlationbased feature

1:21:21

selection and um there's several algorithms that use correlation to

1:21:27

select features but the in in the end of the day the reasoning is

1:21:33

this you want to pick features that are highly

1:21:39

correlated with the class or with the prediction that you want to you want to

1:21:44

uh make and

1:21:50

you you want features

1:21:56

that you want to take features that are highly correlated uh uh to the class and features that are

1:22:05

weekly correlated with each other

1:22:12

meaning that um meaning

1:22:20

that you don't want like the you you don't want the the features to kind of

1:22:26

be uh um repeated carry repeated information so

1:22:34

uh you won't have you you you you want to to have features that are uh not

1:22:40

correlated between them but that carry important information to the class right

1:22:48

to the target so if you take a look at this image here you see that here's a

1:22:54

matrix of correlation so feature one is correlated to the to the target

1:23:01

by.3 so it's a small correlation but feature one and feature

1:23:06

two are very correlated it's 0.7 and even

1:23:14

um um sorry I just read it um the other way so let's go back so the target is

1:23:22

very it's highly correlated with feature three okay so that might be a very good

1:23:27

candidate for for choosing but if you go to feature three

1:23:34

um you will see that feature two is kind of very it's it's kind of correlated

1:23:39

with feature two uh it's not a very very strong

1:23:47

correlation but but you could think about eliminating feature two because

1:23:53

they it's it's highly it's correlated u to feature three

1:24:00

okay um now feature one has low correlation to the target so

1:24:06

uh you know like you you can just eliminate that anyways okay so that's

1:24:13

the the there are several algorithms that recursively goes through this tables and make the selection

1:24:23

so again these are uh these are what we call PCA feature selection they're un

1:24:30

they're uh unsupervised learning algorithms uh feature

1:24:35

selection uh correlationbased it uses it uses

1:24:43

like information from the data but you you're not doing a classification or

1:24:50

regression that's why considered as unsupervised pca is also unsupervised because it it uses the structure the

1:24:58

characteristics the numeric characteristics of the data to come with to come up with the principal components

1:25:05

and so on okay to finish this discussion the discussions on uns in unsupervised

1:25:11

learning uh I just want to discuss again uh what we discuss a new topic named

1:25:19

association rules and this is very common uh association rules is also

1:25:25

unsupervised algorithms that we want

1:25:30

to extract rules from from data right so

1:25:36

we're not doing regression or classification we want to understand what kinds of rules that are

1:25:44

implicit in the data so for example say you have this data set here and um it's

1:25:51

the it's a weather data set where you have attributes outlook temperature humidity windy and play

1:25:59

um whether you should play or not right based on this 1 2 3 4

1:26:06

um um four attributes and what what we want to what

1:26:13

we want to discover is something like this if outlook is equal to overcast

1:26:19

right so when when the weather is overcast you can see here this implies

1:26:28

that you you you can go play and it doesn't matter whatever the other

1:26:34

attributes are right so this is an implication this is a rule and this rule

1:26:40

is kind of uh we extracted it from the table from the data

1:26:46

set or for example and you don't need to you only uh you don't need only to find

1:26:54

rules for the the target for the play right for the for the target you can you

1:27:00

can find rules within different attributes for example if the

1:27:05

temperature is cool so let's go temperature if the temperature is

1:27:11

cool so we have all of this instances you will see that um the humidity is

1:27:20

normal all the time okay so this is also another rule

1:27:28

and if you think about a supermarket data set maybe the person that buys milk always buys um

1:27:36

uh or buys I don't know

1:27:41

um instant um instant chocolate I don't know something like that this kinds of association rules are

1:27:50

very important uh to learn about the data okay so the

1:27:56

way the the algorithms does is it takes

1:28:03

um is it t it takes a what we call an item like it takes a set of attributes

1:28:11

for example humidity wind and and play right and it it tests all the possible

1:28:20

rules um that this three attributes can can um

1:28:28

jointly make right

1:28:39

so okay so I'm not going to explain the algorithm per se but what we want is

1:28:47

rules that jump that that are um that we commonly find in the data set

1:28:58

uh and that are very like that uh that holds most of the time okay so that's

1:29:04

what the algorithms for association rules does it uh this this al the most

1:29:12

common algorithm is called a priori and it's it's a very intelligent

1:29:19

algorithm i can leave a resource if you want to know how this works but in the end of the day it will give

1:29:26

you a number of rules um that are very important for your data

1:29:33

set this is very used in um um it's also

1:29:39

an unsupervised learning algorithm where you're you're kind of extracting from the data this rules and

1:29:47

it's very important for uh marketing and and for uh um co customer

1:29:57

behavior it's it's very very used um to study that okay finally how to uh

1:30:06

evaluate unsupervised learning well there's no way of evaluating

1:30:12

unsupervised learning the common way that we do because we don't have labels

1:30:17

right so for PCA feature selection based in correlation for association rules that's

1:30:26

you know one way of doing it is just applying finding the best features

1:30:32

finding the rules and then see if that modifies the whatever prediction task

1:30:39

you're going to do after that for clustering there is kind of metrics

1:30:47

that we can evaluate our model one of them is being a ratio

1:30:53

between the intracluster distance and the interclusting dist so let's say that you

1:31:00

clusterized clusterized you don't know if this clusters are holds true

1:31:07

um because you don't have labels but one way to measure that is measuring the

1:31:13

distance between points within the same cluster so we want

1:31:19

that that distance between the points in the same cluster to be

1:31:26

low and we also want the points of different clusters to

1:31:33

be far apart so you can you can draw a ratio between the intra the intracluster

1:31:40

um the intercluster over the intracluster distance and this could be a metric to compare for example

1:31:48

different case or different clustering algorithms um there's also another

1:31:59

uh another metric called within cluster sum

1:32:04

of squares where we calculate the sum of squares distance between the data points and their cluster

1:32:10

centers and uh the lower this number the tighter the clusters are

1:32:17

and there's also a a method to evaluate clustering that's called the elbow curve

1:32:24

and the elbow uh the elbow curve it's just because it's it's it's the shape of an elbow where you plot

1:32:33

um the the so you you you for example if you're

1:32:39

using k means you plot uh you have in the x-axis different values of

1:32:45

k and um in the in the yaxis you will have some of this matrix for example the

1:32:52

the intercluster intercluster intercluster ratio and you will just uh you will just

1:33:01

pick the elbow point meaning that this will give uh this will give me the the

1:33:11

best ratio and um again you will never know

1:33:17

the reality but this is a metric to at least compare and and evaluate the

1:33:23

unsupervised learning

1:33:28

algorithms so the this ratio that I'm talking about will be minimum because we

1:33:34

have the intra over inter ratio what we want is the smallest

1:33:42

distance in intra and the biggest um this is the ratio and the biggest inter

1:33:49

distance so the smaller this number the smaller the ratio the bigger this

1:33:55

denominator the smaller the ratio so we're interested in the la in the smallest ratio here

1:34:03

okay all right now just to finish all of this

1:34:09

uh I'm going to I will discuss again in a very shallow way uh what are

1:34:15

recommener systems and I usually put the recommener systems together with

1:34:22

unsupervised learning uh although recommener systems

1:34:28

are very very complex systems uh they are machine you that uses

1:34:36

machine learning models in inside their um inside their inner learnings and

1:34:43

workings but it's recommener systems are complex model complex systems that uses

1:34:51

machine learning but the reason I I kind condensed them together with

1:34:57

unsupervised learning is that one of the most basic recommener systems that you

1:35:02

can do is use clustering so first of all what's a recommener system a recommener

1:35:09

system is a system that makes a recommendation based on data so for

1:35:16

example when you're watching a movie in uh on YouTube and it it recommends the

1:35:21

next video this this recommendation is being

1:35:27

generated by a computer system that that learned at some point from

1:35:35

historical data but not only uh it also learned from the inner structures of the

1:35:42

of the of the data it has so YouTube has data on your past uh his on your history

1:35:49

of movie of of videos watched your behavior where you click or where you

1:35:55

don't where you don't click uh other recommener systems or Netflix

1:36:02

um Amazon all all of this recommendations are are are part of this

1:36:10

systems that we call recommener systems

1:36:15

and um this these recommener systems again

1:36:20

they are they're very complex but the the the most the most archaic I would say method

1:36:29

of doing a recommener system is just by clustering the data so for example if

1:36:35

you have a consumer database a behavioral consumer uh consumer behavioral database you can kind of

1:36:43

clusterize profiles of consumers and then recommend different things for

1:36:50

these different consumers okay so you would you you are using an unsupervised

1:36:56

step of clustering of grouping data and

1:37:01

then after that you're recommending something and this recommendation can be manually just with a knowledge domain so

1:37:09

you know that group A uh would be better off with one offer group B with another

1:37:16

offer group C with another other opt offer but you could also use machine

1:37:21

learning to this next step uh of the recommendation per se

1:37:29

okay but anyways the the most archaic most basic recommener system

1:37:35

is is a clustering right so that's why usually we comment uh and we introduce

1:37:42

recommener systems together with this world of unsupervised learning today

1:37:48

most more modern recommener systems uh are actually based in on

1:37:54

self-supervised learning so recommener systems is a kind of a is a kind of um it has its

1:38:00

own um it's it's a bigger system than just

1:38:05

the the models that we're talking about so it's it's a bigger system and not

1:38:11

just a model uh usually recommener systems you you acquire

1:38:18

data it has a first block um it has a block of pre-processing and

1:38:25

um engineering and and feature kind of

1:38:31

step um then you you do some analysis it

1:38:36

could be it could be a a a first time clustering it could be some kind of

1:38:42

prediction uh you can come up with important features um and then the the fourth step

1:38:52

is using mo machine learning models in what we we call

1:38:59

filtering step it's a what we call a filtering step and that's where several

1:39:05

machine learning models come in uh and it's not as I said not only unsuper

1:39:11

unsupervised such as clustering this would be archaic but it's uh you can use

1:39:18

um uh superi uh supervised um algorithms uh and and most commonly

1:39:27

self-supervised nowadays algorithms on this

1:39:32

filtering step uh which is actually the recommendation uh like the the filtering

1:39:39

step is how the machine learning will come up with the with the recommendation

1:39:45

and and the the difference between the recommener systems resides here on the

1:39:50

filtering step okay so we're going to talk about this

1:39:56

uh uh the filterings and and how and how it works

1:40:03

so the the the way we can interpret the uh the

1:40:09

recommener system uh uh

1:40:16

task is we want to match people

1:40:23

to to recommendations right so uh and

1:40:28

the way to do this is it is to link this

1:40:33

resource is very good i'm going to leave the link here so is to connect

1:40:39

people to for example movies uh let's say D did not necessarily watch the

1:40:47

Matrix um and but it it watched Jurassic Park

1:40:55

it watched others and the the strength of this

1:41:00

connections the the ones that exist u tells you how much the person liked

1:41:06

the movie right so be liked this one more than this one because you see that

1:41:13

this line is thicker so in the end of the day like this is what we want and you can you can

1:41:21

um kind of interpret this this uh

1:41:26

problem as a matrix problem so person A

1:41:32

likes the matrix like four is I like it very much zero is I don't like at all

1:41:38

and so on right so uh so

1:41:43

the recommendation systems is al is is based on

1:41:48

feedback of ratings for example um and

1:41:57

uh one way uh you could you could do

1:42:02

um the recommendations is by as I said uh

1:42:08

clustering so for example person A did not watch um

1:42:15

Jurassic Park but person B did and they kind of if they had similar other

1:42:21

ratings you would just assume that person A will have a

1:42:27

similar um rating for Jurassic Park right so

1:42:32

this is the most archaic way of doing a recommener system but in recommener systems uh we want to

1:42:41

based on the data we have from feedback from ratings or for or for

1:42:48

feedback from questionnaires right oh do you do you enjoy the matrix or do you

1:42:54

enjoy um ad adventure and comedy right you you

1:43:01

will have some kind of relationship between people and objects right

1:43:07

and the recommener system will the the problem that we have at hand is to estimate the values that

1:43:15

we don't have so that we can recommend uh the so we will have to kind of

1:43:23

predict this values or try to figure it out so that we can recommend the most

1:43:30

rated for for person A okay that's the problem

1:43:36

so the first type of um engine to find this missing

1:43:45

values and and to to find the missing values is actually what the filtering

1:43:51

step does the first way to do that the first engine is called

1:43:58

um content filtering okay so let's say that person A

1:44:06

um says that likes comedy a lot not

1:44:11

loves that would be four but it likes comedy a lot but it doesn't like action

1:44:17

okay so this can come this information can come from

1:44:22

questionnaires when you log in for example uh on

1:44:28

Netflix uh they don't use this anymore but they used to do it to ask you this

1:44:34

kinds of questions and at the same time

1:44:41

um Netflix rates or or classifies or not

1:44:47

classif but but characterizes its movies or you could think about Amazon as well

1:44:53

as in in their products as matrix being zero comedy and for action right this is

1:45:01

oversimplified obviously so in the end of the day the calculation of whether person A likes

1:45:09

the matrix uh even if person A did not watch the matrix yet is just based in this in this

1:45:18

features of their preference connecting their preference to to the product and

1:45:25

if you think about I like three times comedy but this has no comedy so it's 3

1:45:31

* 0 I don't like action and this is all action so this is 0 * 4 and in the end

1:45:38

of the day this person is very likely to

1:45:43

not like uh the matrix Okay

1:45:50

so the way the way we do this

1:45:56

um we connect the person to this features and the movies to these

1:46:03

features uh allows me

1:46:08

to to what we call embed into different

1:46:15

vectors this uh of people uh little matrices of people and little matrices

1:46:21

of movies so for example person A likes three times comedy zero

1:46:27

action and remember matrix was was zero comedy and four action and in the end of

1:46:34

the day that matrix that we that we were building is just the multiplication

1:46:42

right so as we did the multiplication for the matrix it will be 3 * uh let me

1:46:50

just go a little bit forward it is 3 * 3 * 0 + 0 * 4 so this is four okay so

1:47:00

that matrix can be constructed by building this what we call an

1:47:05

embedding an embedding is representing preferences

1:47:12

from customers and characteristics of products in in a mathematical way in

1:47:20

different vectors so this is a vector 30 meaning that

1:47:26

um or or the the the or or the the feature comedy is one of the columns and

1:47:34

actions are one of the columns from the person's data set and comedy and action

1:47:40

are also um and here it's flipped right it's it's rows and not columns but just

1:47:47

to uh so that we can mathematically multiply ly but you can think about here

1:47:53

the rows being as well uh the features as well okay so it's just flipped so we can multiply

1:48:00

um and that and that's what we call uh content filtering this is content

1:48:06

filtering so in the end of the day people that have uh very similar uh

1:48:14

characteristics right uh they will have this kind of the same score so think

1:48:20

about B and C they are people that are if you think about clusterization they

1:48:26

are people that are like they're common um this they're not equal but

1:48:33

they they might be part of a cluster right people that love comedy and action

1:48:40

and this like I like comedy and I like action so the

1:48:45

results will be kind of similar for for the for the mult the matrix

1:48:52

multiplication okay so in the end of the day the secret of content filtering is

1:48:59

to create this embeddings okay and in the end of the day if we go back to the

1:49:05

presentation the so the the contentbased filtering is actually the secret is in

1:49:12

kind of doing this embedded uh for example if we're representing

1:49:19

uh here's a two-dimensional graph we're representing comedy and I don't know ad

1:49:26

um comedy and action men in Black is closer to

1:49:33

um Arrival for example right so oh I don't have Arrival here but Men in Black

1:49:40

is 04 it's closer to 13 right to Jurassic

1:49:46

Park than to Nhole 40 so if you if you

1:49:52

think about this representation this vector is in this features space

1:50:01

okay they are closer together so and you will do this for the products

1:50:08

or films and also for the users who are like you will see that people are more

1:50:15

similar than than than the others will have similar results okay so this is

1:50:21

what we call embedding uh is this step of kind of

1:50:26

generating vectors for for the features of of the movies and users this

1:50:37

let's talk about collaborative filtering um and the diff so content

1:50:43

filtering is basically uh a human kind of um

1:50:49

uh getting information from you and creating this embedding in known uh uh

1:50:56

known features right so kind of creating vectors in this space of attributes that

1:51:03

represent people and products right and usually you can do that for

1:51:10

example you have several movies uh and you can uh as as the simplest way is

1:51:18

kind of um um give give scores on each

1:51:24

uh label here of the each variable here of the of the feature space the other

1:51:31

way around is to uh like

1:51:38

clusterized clusterize on uh actors on uh tags and labels that that

1:51:49

people give to this to this movie so there are more

1:51:55

um complex ways to do it uh but in the end of the day that's the

1:52:00

the reasoning so the other way of filtering is what we call collaborative

1:52:07

filtering and the uh the concept of collaborate filtering is this why don't

1:52:13

we use a machine

1:52:19

learning to do the trick of the embedding as well right so

1:52:25

uh let's say that we have the

1:52:31

matrix and we have the missing value so A didn't watch Jurassic Park so I'm not

1:52:37

sure if I should recommend Jurassic Park or or should I recommend airplane airplanes so I have the the

1:52:46

problem is continues to be the same i have to kind of predict this uh this

1:52:52

scores but now I will use a machine learning

1:52:57

to to actually with the data that I have with the data that I have I want I will

1:53:05

approximate the embedding so both of these matrices I will learn this

1:53:13

matrices right to approximately learn

1:53:20

um I will learn matrix of people so the embeddings of the people and the

1:53:26

embeddings of the movies uh in order to when I multiply this

1:53:32

matrices it's it's closest as it can be to the data points that I have right so

1:53:41

the the way you do this using machine learning is you start random random

1:53:46

random matrices for products and users and then you keep updating as a a way of

1:53:55

approximating this the numbers that you have right when you achieve this

1:54:03

approximation then you have you you stop the algorithm when you achieve a local

1:54:08

uh a minima uh you stop this approximation so now if you want to

1:54:15

predict person A to Jurassic Park you again will just multiply 0 * 3 2 * 0 and

1:54:23

that's zero but that doesn't mean that this column this feature is comedy and

1:54:30

this is um um I don't know what it was here but

1:54:37

this this might be not what a human would think it is so

1:54:46

this is the beauty of using content uh collaborative filtering is that the

1:54:52

important features of the embeddings are actually whatever the machine think it's

1:54:58

important we don't have um we don't know what the features are right um so this

1:55:08

is a if you go back to week one you will see that this is kind of a self-supervised learning so I'm in the

1:55:14

end of the day my algorith algorithm is having some data it's approximating generating what

1:55:21

we call pseudo labels or what we call a latent space or latent features meaning

1:55:27

that they are important but I don't know what exactly they are the machine kind of just made

1:55:34

these vectors and it's just repres representing as best as it can the

1:55:41

matrix multiplication right um so the difference between content

1:55:48

filtering is the content filtering you

1:55:53

do you can use machine learning uh in terms of clusterizing grouping movies grouping

1:56:02

people uh for example behavioral data um

1:56:08

and do the recommendations but based on this features that are known from your

1:56:14

you know the features that you're talking about in terms of the

1:56:20

embedding and in terms of that little matrices but at the the con the

1:56:26

collaborative filter filtering is something more mystic like the machine learning

1:56:31

is actually learning um the is actually learning the data right

1:56:39

it's it's actually learning important aspects of the data that maybe we're not

1:56:44

perceiving right there is there are works that that after the machine

1:56:53

learning learns oh sorry the embeddings after the machine learn

1:57:00

machine learning learned embeddings we can kind of analyze this we could you

1:57:06

know like after the learning we might come here with some kind of expert in in

1:57:13

cinema for example and kind of identify I that Oh this was talking about

1:57:20

um um short

1:57:25

movies this is comedy the other one is

1:57:30

uh contains classical classic music there's several several several

1:57:36

different things uh that the that that the machine learning can

1:57:42

perceive that we cannot perceive uh just by looking at this numbers like

1:57:48

the the machine is well uh is way better than us to do that um so just to wrap up so so how to

1:57:57

evaluate uh recommener systems uh there are

1:58:03

several metrics um precision what we call like in your

1:58:10

recommendations how much you've you've got it right so for example I

1:58:17

recommended six items and the person so you you hold out

1:58:23

a purchase that you can compare and uh and you test your recommendation against that so I got one

1:58:31

two three four items out of six this is called precision is within your

1:58:37

recommendations how much you got right and the other is recall um is

1:58:45

um the the the it's it's the other way around so within the

1:58:51

purchases that the person did how many you got it right

1:58:58

so you got it right uh

1:59:04

one two

1:59:10

three and this one too right so um just one second so within Oh yeah so with so

1:59:18

within the purchases how many the recommendation got it right so one this it got right

1:59:25

this got it right so I have to to look at this row within the purchases this

1:59:30

this this and that so it's four over five and what we want is to maximize

1:59:36

precision and maximize recall so there are some

1:59:42

uh we want this recommendations and that's the way uh you evaluate those

1:59:47

recommendations again recommener system is so so complex there are so many

1:59:54

algorithms and machine learning ways to build recommener systems to build this

2:00:00

filterings to build to build to build clustering embeddings it's it's a niche

2:00:08

it's a very um complex niche that uses lots of machine learning and all types

2:00:15

of machine learning into it but because we just as as we said the most pre

2:00:22

archaic type of uh learning for recommener system is

2:00:29

is clustering kind of grouping uh also

2:00:35

like choosing features that are important right so if you think about

2:00:41

doing contentbased filtering you could maybe find uh the the the ones that are

2:00:47

most important right to the the features that are most important anyway so uh you

2:00:54

would do that in the analysis step here uh and then uh you would filter that

2:01:02

using the content space but you can also use machine learning to learn the latent

2:01:08

space uh and it's it's a very complex world so this is this is not meant to be

2:01:14

um um by any means a more in-depth uh

2:01:19

discussion but a very shallow introduction okay so for this week I

2:01:27

foresee three three prototypes um one on clustering one on

2:01:34

dimensionality reduction and association rules and the other one in recommener

2:01:40

systems but a very basic one again this is very complex uh just using

2:01:46

clustering and and um feature selection and and kind of you

2:01:52

know be a very basic recommener system um I already kind of picked some

2:01:59

examples uh and I'll I'll try to work them out i'm I'm I'm I'm outlining this

2:02:04

the examples as we finish the content for each week uh I'm outlining it but

2:02:11

then after we finish the content for week five then I'm going to start unfolding uh unfolding the

2:02:18

prototypes all right cool

English (auto-generated)