You have. I. Good morning. Now you can actually hear me. Thank you. Microphone for working with me. Good morning everyone. I see we've we've lost quite a few people. I'll take that as people doing the right choice because I hear some chattering about people getting sick. And if you show up in my lecture and you're sick, I will be very angry. Please stay at home. Please don't infect all the people around you and don't infect me. I do not have time to get sick right now. I'm very busy. So please, if you're sick, stay at home. No one will be mad. There will be time to catch up on the material. Lectures are being recorded so you can watch them from the comfort of your home with a cup of tea in your hands after you've covered. Yes. So we at week eight, we are nearing the end of this course. As you know, there's only 11 weeks of material, so we have eight, nine, ten, 11 formal lectures to go through. And we're kind of now in the middle of talking about. Talking about different methods. So we'll have a new method covered each lecture. And today's lecture will be relatively relaxed to kind of prep you for next week, which will be a bit more intense and a bit longer. So we'll have some time for questions at the end, hopefully. So we might go through some of the material we have covered so far through the lectures, so it will be a bit more relaxed this time, don't worry. But first of all, I would like to thank you again for attending last week's guest lecture. I hope you enjoyed it. Carlos taught me again afterwards that he he really, really enjoyed it. It was a new lecture for him, so he has given guest lectures in the past, but he kind of re revamped all his material and all his slides and kind of the content of his lecture. So he asked me to go on various things for you, to you for attending and asking good questions and listening to him. And he also asked me to please tell you that if you have questions or feedback, he would be really interested in hearing that. So I'm happy to forward his contact details. I think you can find him on LinkedIn. He was very keen to connect with all of you on LinkedIn, so don't be shy to reach out to him if you're interested. And and I also wanted to take this moment to maybe collect a bit of comments or feedback from you. If you would rather be anonymous, or you would rather share it with the class. So any any thoughts, any feedback on the guest lecture? What, what? Well, what didn't want. What didn't go well, what did you enjoy? Was it good overall? Any thoughts at all? Yes, I quite enjoyed it. I thought he. Really? Well, I like this explanation of the modelling. Good. Yes. If you enjoyed it. I also enjoyed the fact that it's not easy to get an insider perspective when you are not there. So it was nice to know how a plan works, how they how their thought process works. Yeah, I thought it was really interesting to hear how long the process on the one hand, how long, on the other hand, how short the modelling process was. So I think it mentioned something about six months from beginning of talking about something to actually getting the results, which on the one hand sounds quite long, on the other it is roughly how long. For example, an undergraduate dissertation I think is one year, but we usually say around six months. So you will have half the time, I think, for your dissertation, which might also tell you something about how big or how small the project might be for your dissertation. So I thought it was really interesting to hear more about the timeline and the steps they have to go through. Obviously, in a company, what you have to take into account is there's a lot of a lot more regulation and steps and convincing people and all of that. You might have actually experienced that more during your product group work, where you had to convince your group members that something is the right thing to do. And I see glances being exchanged where arguments were being held. Any other feedback or comments? Generally happy. So something we should recommend and keep in the curriculum and do again next year. Very good. So I will let him know and I will also. If you have any more feedback, please do let me know and I will forward that. And I'm sure a bit of criticism wouldn't hurt him either. And the other thing I wanted to talk about is I've been asked multiple times about the exam timetable. In my defence, I do not create exam timetables. So we have a central timetabling unit in the university. As you can imagine, scheduling scheduling exams for a huge university such as ours is an extremely difficult and complicated task. I do wonder sometimes whether they reach out to us in regards to optimisation methods that we could implement. Maybe that is something we should offer, but no. And so yes, the third thing is that timetable is online. Now this is the link to that. So please check the timetable yourself in case there's any changes. And then please don't use me as the excuse if the timetable has been if the exam has been rescheduled. But my lecture was said it was this at that time, it's your responsibility to please check it yourself. But the last time I checked it was the 18th of December. So you're very lucky with that. I think we have exams until the 23rd. So you might have spent half of Christmas here at 2 p.m. to 4 p.m.. I'm not entirely sure where that is, to be honest. Thomson's land, I think it somewhere by Porlock holds. I know. Has anyone checked what it is? It is in Edinburgh. It's very central. So it should be. Should be walkable. Just like any other location is as well. So this is the main exam for everyone. If you have any special arrangements, obviously check please with your personalised timetable. I think you can access that via the same link and if you have any kind of accommodations then you will be informed of your own exam. You might have for example, more a longer time than that. Then in that case you will likely be in the same main example, but you will have extended time to write your exam towards the end. Any questions about the exam? I've uploaded a sample exam from last year to learn, so please, please have a look at that. And I think that will give you a good idea of what the exam will look like. So I've been asked whether there will be complicated calculations. No, there won't be calculations unless you want to calculate an example in explaining a concept, and I'm very happy with that as well. Otherwise it will be similar to last year's exam. Obviously not the same, but similar. Okay. Let's go into our recap. So our last lecture was actually in week five. So then we had reading week and six and guest lecturer in seven. So it's been a while. But last time in week five we had a bit of a recap around data splitting, resampling, class imbalance handling is all of these things that we've also covered in last week's tutorial. And you'll remember, data splitting was this idea of dividing your data into training and test data, and then just training your model on one part of the data and testing and all the other cross-validation and we sampling was this idea of repeatedly training a model, repeatedly running the model on parts of your data. And when we were talking about class imbalance handling, this was the idea of if you have very imbalanced classes, especially if that is your outcome variable. So it's much less of a problem if it's one of your predictors. Sometimes it is a problem, but usually if it's one of your predictors, you don't really care too much about class imbalance. But if you're trying to predict a class and one of the classes is much more frequent than the other, then the model will naturally adapt more to the more frequent class. So in those cases we often use imbalance. Yeah balancing methods kind of upsampling downsampling smote all of these to even out the classes. Oh gosh. We were then talking about unsupervised machine learning. So we were talking about partitioning clustering algorithms. That was your K means your K meteoroids. And we were talking about hierarchical clustering where we're creating a dent. So basically a structure of multiple clustering solutions instead of a single solution. Now this week we will start by talking about k and n which is one of our nearest neighbour classification approaches. So we are leaving unsupervised learning and we're going into classification now. And we will also talk about feature selection, dimensionality reduction. Specifically I will talk briefly about factor analysis. But then the majority of the time we will spend on principal component analysis at PCA and regression. So let's start talking about canon. And I put purposefully into this lecture because it is kind of a natural connection between clustering, which we've covered in week five, and the classification which we'll cover in the coming weeks, because the idea is extremely similar. So it will sound a bit like K-means to you. We are still trying to find patterns, and in this case, we're trying to find classes of points by looking for some kind of similarity or closeness between data points. And we still assume that points which are in the same group or last last time it was cluster behaves somewhat similarly in terms of specific variables. So this is the idea of clustering. And it's also the idea of a nearest neighbour classification. The major difference between this is that in this case, we still have new and unclassified data points, and we're trying to assign them to some group or cluster or class, and we do so by analysing the neighbours. However, the big difference is that in this case this is a classification approach. So it's a supervised approach which makes it different from clustering, where we just basically took all of our data and threw it at the algorithm and hope that the algorithm produced something. So we were talking about how clustering has this big disadvantage that you can't or it's very difficult to control what kind of result you get out of it. It's very well unsupervised. It just does it does its thing and you hope that the result will be useful for you. There are some methods where you can then choose the number of clusters, or you can kind of optimise regarding a specific criterion or linkage or all of these things where you can kind of lead the algorithm to a good solution. But what defines a good solution is extremely difficult to say in a clustering problem. So I sometimes get asked, well, what is a good result from clustering? And it's almost impossible to tell. You can kind of use evaluation criteria where you look at for example how distinct are the groups, how well separated are the groups? Are they offer good and equal size? All of these things. But it doesn't really tell you a concrete kind of number as in how many elements are in the correct cluster, because there's no such thing as a correct cluster. You don't really know which group each of these points should belong to. Now, in supervised learning, we know what the correct classification should be, at least for our training data. So we have some kind of pre labelled training data where we have a selection of data points and we know exactly which group they belong to. And that is our answer for each of these. And our goal is now to classify new groups, new members that we don't have previously labelled and assign them to the same already existing classes that we know of. So in this case, we can actually see how well does our algorithm perform in terms of classification result. So very similar to for example logistic regression where we can count the elements which were correctly classified. Okay. And this is probably the most popular neighbour classification approaches. I would struggle telling you another one. I know there are different neighbour classification approaches, but Kagan is so dominant in this area that it's actually difficult to remember any of those because there's always killing, or at least it's something that sounds like an N and should be called N, but got a new name to sound more fancy or new, anything like that. So K and then works in four relatively simple steps. We first have our training data and that is some feature combination. So some variables and a class label. So you have some kind of x and some kind of y that you're trying to predict the same as you do for your logistic regression. And then for any new and unlabelled data points that you have in the data, you look at the k nearest neighbour points surrounding that data point and which class they belong to. Now, in a two dimensional space that is relatively easy to imagine because it's basically just if you have a bunch of points and you look at one point, you don't know which class it is, which class are the points around it. It's very simple logic. So obviously you have to define what does close mean again. So we're coming back to this idea of dissimilarity measures distance measures. And this is another connection that we have to clustering. It's the same idea really. So depending on your application you might for example use a Euclidean distance, define the distance of your unlabelled points to all of its neighbours, and then look at the k nearest of those. So you order them in decreasing and decreasing order of of closeness. And then you basically look at the k ones which are the closest to it. Then they just do a majority vote. So if there are three points, three nearest neighbours and two of them say class one, one of them says class two, then you assign it to class one. So this is an example of what that might look like. Here you can actually also see the impact it has depending on how many neighbours you choose for your data point. So you can see here in green our data point that we have not labelled yet. We don't know which class this point should belong to. And we have two classes, the blue squares and the red triangles. And the question is which of which class should our green point be assigned to? Now the inner circle here shows you what happens if you decide that your k your nearest neighbours should be three. So we look at the three nearest points that we have here, and we can see we have two red triangles and one blue square. So obviously if these three points vote then we would assign our green point the label red triangle. Now what happens if we actually decide that our case should be five. So in other words we look in a wider distance radius a wider radius around the point. And suddenly our votes change because suddenly we have three votes for no, there should be a blue square, and only two votes for this should be a red triangle. So in this case this would become a blue square. And this already tells you that the most tricky thing about k-NN is the same thing that we already struggled with. With k means you have to choose your K, and I told you last time that K would become your least favourite parameter name for that very reason, because you always have to struggle choosing it and has a really big impact on your results. So the choice of cake can be extremely challenging for the basic reason that if you choose a two to large, you're basically smoothing over any small dynamics that you might have. So in this example here, if you choose a really, really big cake that kind of encompasses all of these data points, then this kind of small dynamic here with the red triangles in the middle would be completely smoothed over. You would not notice it at all, but if you choose k really small like k three earlier, then you might assign this point to the red triangles, even though if you look at the overall picture, maybe a blue square would not be a bad fit for this point. This also is a bit tricky because I said earlier, K and is so easy because you can actually calculate whether a point is correctly classified or not. But when we say correctly classified or not, this all depends on your choice of neighbours. So what does correct classified for this point really mean? It could be, hey, we want to pick up this kind of small dynamics because we're interested in these smaller group behaviours. We're interested in subgroups, for example of consumers or anything like that. Or it could be no, actually. What kind of prefer a more broad view. We're not really interested in these outliers or individuals within that. So that's not really a correct classification until you choose your K, or at least until you decide what your desired business outcome would be, or how you would like to approach this problem from a business perspective. So, yes. Too large. You're smoothing over. Too small. You're sensitive to any outliers. So if you have, for example, a singular red triangle anywhere, then you would assign a point maybe to that group, even though it's just an outlier at the very edge of a much larger blue square group. So as K impacts your modelling results, you can actually be tuned and therefore we can adapt this whole problem to very different application contexts. And we can therefore call the tuning parameter because we can kind of change it in tune depending on what we prefer for our application. Now in practice. And I do include our our computer lab in that tomorrow we will actually use cross-validation and we will test multiple values of k. And then you can optimise that in terms of some kind of classification accuracy. So if you know in your training data you have all your data pre labelled. So you know with definite accuracy your your label for each of these points. You can therefore use cross-validation test multiple values of k on your training data where you know your accuracy and then you use that k on your test data. So it is a very similar approach to k means you remember that in k means we were talking about different ways of optimising and finding the best number of k, and we were talking about this elbow criterion. We were talking about silhouette score and all of these different measures. So K and as a supervised problem makes it a little easier because it just makes it easier to quantify performance of the algorithm, because we can actually calculate the accuracy on the training data. The other consideration that I think is really important for Ken, and I briefly mentioned it earlier, is the choice of distance measure is really crucial. So I think a lot of people make the mistake that they just throw Euclidean distance at their data and hope it works out, and in some cases it works just fine. If you have all numeric data and it's all very pretty. And by all means, Euclidean distance is the way to go. And I think it's also the distance we will use in our example in the computer lab, because we're just looking at two dimensions of numeric data, clearly Euclidean distance, and be done with it. Don't overthink it. But if you have mixed data and the social science, we have that a lot and suddenly it becomes much more difficult. So if you think back to our lecture on clustering algorithms, we were talking about these different distance measures that we can use depending on our context. So we had some that were better for mixed data types or for ordinal data or kind of just one data type which is numeric and all of that. So all of these things preprocessing, scaling your variables really, really important for k and n and then choosing a distance measure. All of that is extremely important to consider before even starting to think about what is the optimal number of neighbours. So yes, different data calls for different ways of measuring distance. I think I had a bit of a rant about time series data and spatial data. What even does distance mean? Which becomes a bit of a philosophical question. So if you want to think about that some more than please do, I sometimes feel like I'm the only person in the world who thinks, what does distance even mean in different contexts? Most people are just throw some method at it and it works out for them. So I guess that's fine. I guess that's what you can do. Yes, really interesting as well. In some cases we assign a weight to neighbours which is proportionate to the distance. So this can reduce the impact of outliers if they're really far away. So basically instead of just looking at the number of neighbours that you have. So for example in our earlier example here, this neighbour here, the red triangle is so close to the to to our unknown, unknown point that we would say this is a really important point in determining the class of our kind of unlabelled point. Where is this point very far away? We might say it's so far away that it doesn't really have much impact on the class that it should have. So there is a way of kind of weighting the distances, which would make it a bit less sensitive to outliers. And it can also help in if you have very unevenly distributed density of points, for example. So one of my research areas is actually thinking about how density of points determines what classifies a cluster in my case. So if you think about it that in clustering, if you're interested in finding groups of data points in a specific to the international region can be geospatial. Doesn't have to be whether something is a cluster or a group or points that belong together, which is k and n depends on not only kind of where something is located, but what points are surrounding it, and how densely are these points clustered together compared to what points look like around it. So it's a really interesting concept that in areas where there's less density, for example, a cluster isn't necessarily the super dense area of points, because naturally the points are less dense in that area. So this is a related related concept where we're looking at how the distance interplays with the number of neighbours that we would like to consider for assigning a membership to our new point. Yes. So sometimes you assign a way to neighbours which is proportionate to the distance. Last but not least, then unfortunately, because there's always an unfortunately that's not perform very well in high dimensional spaces. So k and n is works really well if you have like a couple of variables, a couple of dimensions, but it's very difficult to use it in high dimensional cases. So with a lot of features, a lot of variables. And part of that reason is if you think back to this idea of calculating the distance to the neighbours, if you have a lot of data points, that's already becoming difficult, because for each of these new data points, you would have to calculate the distance to each other data point in the whole data set in order to determine who is actually closest to them, because you don't know that yet. You actually have to calculate each of these distances. Now the more dimension that has, the more variables that has the more complex this distance calculation can get and the more computationally expensive it is. So Kanan is amazing for small to medium sized data sets, but as soon as you scale up to large data, a lot of dimensions, it really struggles. And it mostly struggles because of this idea of calculating distance to each of your neighbours. So there are some ways of about that. Theoretically, you could, for example, combine it with different approaches so you could divide your space in some measure. You could use clustering for example, to divide your space into subspaces. And then within these subspaces you could use k and n because that would restrict how far away you actually looking for neighbours to your points. So there are some ways of overcoming that. But typically we wouldn't use k n only in low dimensional cases and typically in small to medium sized data sets. Now I'm extremely proud of this pun. So thinking back both on clustering and on k and N, you can discuss with your nearest neighbours and K is your choice and some of the advantages and disadvantages of clustering, because we did not come around to talking about that after our election, I think it's really important. Some advantages and disadvantages of k and N, which we just discussed. You can contrast it to clustering if you like. And then I think this is an interesting one. When do you think you would choose one or the other. So when do you think clustering approach any clustering approach would be more suitable for your data. Or K and n would be more suitable for your data. Okay. So take a couple of minutes, maybe ten to talk to your neighbours and then be collect. Should we collect a couple of points together? Okay. Should we start? Maybe. What do we say? Advantages and disadvantages of clustering. And then advantages and disadvantages of k and n. Maybe in contrast to that. And then when would you choose one or the other. So let's do maybe a couple of vantages. You know I should really is it. I should not do it that way. My handwriting is terrible. My handwriting isn't good in the best of times. This. Yes. Much better. I'm sure you're aware that the exam in December is is handwritten, like, right, like piece of paper and a pencil. You know what a pencil is? Yes. So I would suggest that you start practising maybe writing notes by hand, because I know that if you're out of practice and you suddenly have to write for two hours, it will paint your hands. So start practising now. I remember last year a couple of students were complaining about that it hurt and that they are not used to writing by hand. So don't say I didn't warn you. Okay? That being said, I don't knock you down for your handwriting. Yes. I was actually I saw that there are different marks for question for assignments. So how many pages are. Like how exactly are we supposed to structure. I don't know whether it will be the same point distribution this time that me. So that's the first thing. Generally, what I would suggest is the whole the total marks available at 100. And you know, there's two hours total allocated. So that way you can kind of calculate how many minutes, how much time you should spend roughly on how many points. So if 100 points two hours then an hour per 100 points ish maybe. So that way you can kind of calculate down how much space or time you should spend per question. I don't have any thoughts on how long a good answer should be. Depends completely on you. I have a well thought out, concise answer is equally good as a long answer, as long as they cover both the same points. Yeah. What is like the meeting criteria like in the 35 mark question? I work with the marking criteria. You will be like is there? Yeah. So the way I construct an exam is I have an idea of what I would like to read. So that's basically what kind of points should you mention to to kind of get. Yeah. Get full marks. And then there are obviously points for things like critical thinking and discussion skills and details and all of that. So it's more of a how would you say that holistic holistic view. But I think it's comparable to the coursework marking the main differences. I don't expect you to do research, obviously. So there's no marks for remembering book titles or anything like that. But the idea of critical thinking and discussion and then I have a list of main points which I would like to read. Okay. Yeah. Let's look at this lens. Big enough. What are we expected to do? Some extra reading on the docket. That's a really difficult question, because the you will notice that the way these lectures are structured is that there is an emphasis on understanding a concept or a technique for kind of from an overall perspective and then its application. So how you understand what the technique does doesn't really matter to me. If you find my, my explanations intuitive and you get what the method achieves, how, then that's fine. And you can you can use that if you prefer the textbook, because it explains things better than I do. That's fine. If you like a different textbook, that's fine. I just would like you to get into this habit of understanding. How different methods work, because that allows you to choose the right method for the right problem. So if you go back to our learning objectives, I think that is basically the best thing you can do. It's also the best thing you can do if you're wondering about marking criteria, because the learning objectives are what guides me in constructing a course and constructing an essay or an exam. Okay, we'll have a bit more time at the end for further exam questions, so don't worry. But let's focus on our friend clustering for now. My favourite. So what are some advantages or disadvantages of clustering which is you've discussed. Yes. To kind of discover patterns or relationships in your data. Yeah. So there's a big focus on discovery basically. So patterns, relationships all of that. It's all about this kind of explorative perspective on data. Yes. Very good. Another advantage or disadvantage. It was too long ago. Yes. It's like. Like using clustering. Interesting. So basically use clustering to to detect outliers. Just like you can clustering. Deviation on the groups and discovering the means for each group. How would use your outside? Ah. So. Ah. So basically because instead of looking at individual data points and outliers, you're looking at the structure overall or the group instead of the individual, which would basically. That's a good point. I think there are some methods which are sadly very sensitive to outliers. So we were talking about **hierarchical clustering.** Hierarchical clustering is notorious for that. I struggle using it every time because it tends to really grab these outliers very much out there. So hierarchical clustering is quite sensitive to outliers, especially if you use, for example, single linkage as your linkage criterion. It's better if you use complete linkage, but still it's quite sensitive. But you're right that in kind of the perspective on analysis. So you're basically grouping together data points. So instead of having the impact of each of your data points on your interpretation, for example, you describe in groups, there's one of the reasons why clustering is used in market research market segmentation where you're then describing groups of class of customers, for example, through personas. So if any of you have a background in marketing, you'll know this concept of a persona which is a basically personified costumer and artificial costumer who you construct using criterion from, for example, clustering. So you would look at the average person of a cluster and construct a persona out of that, which helps you in creating marketing market communication tools. And how do I best ride that? So it's maybe role of the individual data point versus group, point versus group. And then I put outliers and in quest and quotation marks and you'll all remember what I said. With my words, and you'll be able to remember my words better than my written notes. Hopefully. Yes. Anything else? Any disadvantages? Yes I was. I think the difference between clustering and a black box model is with clustering. You do know why group points are grouped together, so you understand that points are grouped together based on the distance to each other. For example, it's kind of more intuitive. Whereas with a black box model, you really have no idea why a specific point is assigned to a specific specific group. So there's a difference between the two. But I think you're touching upon a good point because it is. One of the disadvantages is it's by nature unsupervised. So you can't really you can't really steer it into a, into a direction you can't influence or steer it really. Which can be an advantage because sometimes you just don't know what's happening in your data. So you just want something that works. But sometimes you have a specific idea of what should be in the data, and you're trying clustering and just giving you something else. And it can be very frustrating. Any other advantages disadvantages or would you like to move on to N. Yes. Like I said, bias does I have. Um, does that mean the more data you introduce? I mean with. Yes. So you're referring to this bias variance trade-off thing whether you're overfitting basically. I mean, the the good thing about clustering is you're already giving it all the data. So there's not really additional data that you might be overfitting against versus with classification. Sometimes you overfit to the data and the new data can't really be classified. That being said, with clustering, I think we talked about what happens if you have very kind of small clusters, small groups, you're overemphasising small patterns. And in that case, yes, it's basically overfitting because if you do create new data, new customers, for example, then it can be difficult for that method to run. And you will basically have to restart. We start the method instead of assigning points back to the clusters. I mean, you can do a hybrid approach. You could do clustering first and then assign new points using k and n to the already existing clusters. But then that might lead to overfitting. In that case. Yes. So do let's do put that here. So might overemphasise patterns which means not generalisable to new data. Emphasise. Oh, okay. No, it just doesn't like my British English, I think. Okay. Okay. And then. Yes. You don't need a training phase for men. I mean yeah you can't. You don't have to. Yes. So it's basically what we usually would do is we would tune it towards K. So the number the number of number of clusters k. Yes. But then it's a bit of A11 off method. You have k and you basically throw it in. So yes. So there is more less of a reliance on training I'd say. And. So we'll, we'll see that in the computer lab. What we mean by that compared to for example, neural networks which take many, many rounds of tuning, not only the hyperparameters but each of the individual parameters within the model itself versus k and n, which is a bit quicker in that. Yes. Anything else on KNM? Yeah. Choosing the cake. Yeah. And I mean, we did have that earlier with choosing K. We did have that with clustering, which we can put as another disadvantage to clustering is the same problem of. As soon as you have any type of input from the researcher you get, you can get into a bit of trouble. So for example, with k means you have to choose k and a distance measure. With your hierarchical clustering you have to choose a distance measure and the linkage measure. So in all of these cases, choosing that in a way that fits the data best can be tricky. And with k and N it's the same thing. We had that earlier when we were talking about this example with the squares and the triangles. When we're talking about the role of k and what even is an optimal k, it depends on your application. So yes choosing k tricky. Yeah. On the large data points. Yeah we were talking about that. Large and high dimensions. Now. Later. So yes, it suffers from that. Definitely. There are methods which work much better on large data sets. There are some which I mean it can be an advantage as well. It works really well on small data, which is something we're talking about. Logistic regression. Logistic regression works really well on small data and k and works well on small data. But then there are some methods which work just better or only on large. So neural networks work only on large data sets. Support vector machines work really well on large data sets. Random forests great for large datasets? Yes. Mm. I don't know. If this is a disadvantage, but don't we have to standardise data when we use GNN? Yeah. So k and n is scale sensitive and that is that that is due to its use of distance measures. So in order to define what a distance is between two points you have to standardise the data, because otherwise one of the scales will kind of skew the whole. Without distance measure distance. If you have purely numeric data beautiful basic numeric data, standardise it by with Euclidean distance be done with it. Yeah. But otherwise if you have mixed data depends. You can use things like the Shaka coefficient on all of these beautiful names that we learned. Yeah. I want to ask question. When we can consider a dataset. Depends on how good your computer is. No, it really depends. So what we considered a large data set in the past is not large anymore. So nowadays what I would consider a small data set is a couple hundred two or couple maybe two 3000 data points. Medium for me would be in the thousands to maybe 10,000, and then large would be 100,000 to millions of rows. But that's a rule of thumb. So that's based on my laptop. And if you have a supercomputer, suddenly everything is very small. I think we had we had one more point here, right? Yeah. You can use a larger dataset. It depends on the method. Yes. So k means for example, is really computationally efficient. So it works well on large data sets. Hierarchical clustering is a bit more expensive. So it depends on the algorithm. But generally clustering works on medium data sets. At the very least some of the methods on large data sets as well. To can. And you said it works better on smaller datasets. But if the dataset is too small. Like is the issue. With large datasets the fact that it's computationally expensive? Yeah. To be more accurate if you have to. Yeah. So the problem with large datasets and then is purely computationally so large datasets, because we have to calculate the distance between each of these points Paris gets too expensive. You're absolutely right. If a dataset is too small then obviously how much does it actually tell us about the data. So there's always this this balance to strike. So if you only have 20 data points, are you really discovering rules of classification with 20 data points. On the other hand, you remember back in the principal's lecture we were talking about what is small and large data depends on how large your population is. So a sample can be large. In regards to its population, even though it's small. If you only want to research a really, really small group of people, then your sample might be tiny, but it's still showing a significant proportion of the whole population. In that case, a small data set can be sufficient. So we think about the population that you're actually trying to measure, and how well are you able to generalise from your sample to that. And the advantages of Kaden. You apparently don't like it. Yeah. I think that the computational time is a bit less compared to the. Yeah, it's relatively easy, which means it's I mean, for small and medium data sets, it's computationally relatively efficient. So there's no kind of going through all of these training sizes. So can be computationally P. Oh, God. Computationally efficient. And. And it's relatively. Easy to understand and interpret, which I think is related to that. So if we were talking about communication, explaining what your model does, we're talking about black box models earlier. All of that isn't too much a problem with. Can I explain to you in in 15 minutes. And I think you got the idea. You might not get every single detail, but you got the ideas. It's really intuitive. It's something that you can explain well to a manager. You can basically explain, well, this point is similar to other points and therefore it's in the same group. That's it. That's really the idea of k and n. So compared to a neural network, which sounds really fancy and nice, how do you actually explain what it does. It's very tricky. We'll spend a whole lecture on that, not just half. Okay. Yes, I think maybe. Because that purchasing the centroids. For Cain or for. Choosing the centroids. When you calculate the distance. No we don't. We don't really calculate the centroids. So what we would basically do is we would calculate for this green point the distance to this one and this one and this one. So all of these points. And then we would choose the three smallest distances from that. So we don't really calculate a centroid as we would with k and n with K-means. With K-means we had this cluster centre point from where we calculated all the distances. And then we move that around iteratively. So with k and n we just calculate all the point pair distances. Yeah. Huh? So the central point is just collected. Selected randomly. With the number of k. With k means or with K9 k, then we don't really have a centre point with K, and that's a new data point which is unclassified. It looks like a centre point, but it's not. It's just a new data point which we basically plug in there and we want to know is it a red one or is it a blue one? It looks like a centre. Point, right? I think the reason why it looks like a centre point is kind of the the circles around it, but the circles around it are just to show you how we look for neighbours. So the small circle is k three and the large circle is k five. It's just a search radius around that new data point where we look for points. Yeah. Another question. Mm. The observation, can users. Have to be labelled? Can be labelled. And though the label is a class so it's not a regression problem, the label for it has to be a class label. So it can be a string or a number, but it has to be like a label. For example, good customer or bad customer. Now it's not a regression problem. So in regression we are trying to predict a continuous number and classification we predict a label. Yeah. Yeah. Yes. We can have a break. And then after the break, we'll briefly talk about continual discussion about a comparison. And then we'll do a bit of feature selection. Okay. So five, ten minutes. Yes. Let's. I can. Welcome back, welcome back. Listen to me, I am talking. To you. Okay? So I hope you had a good break. I got a couple of. I got a question on an intuitive example for K and how it might be used, especially in comparison to clustering, which I thought might be quite interesting or quite helpful for you. So I'm sorry to always bring out these marketing examples. It's definitely my background. It's why I feel most comfortable. So you have to deal with it. But I'm sure you're able to to adapt that to your own interests and your own background. So if you imagine your company like a bank, like Carlos was talking about, I voted last time and he was talking about how different groups of customers should be targeted with different types of communication. So for example, a mortgage offer or something like that at different points in time through different channels. So in order to achieve that, you have to understand your different customer groups within your database. And there are different ways of finding that out and achieving that. If you have absolutely no idea what type of what kind of different customers you have in your database, then clustering would be definitely the way to go. So you could, for example, use a number of dimensions on customer behaviour like their income, their gender, their age, their location, I don't know, something like that. And use those as dimensions for your clustering and you would create different groups of customers according to those dimensions. Now you can then go ahead and describe these groups in terms of the averages. But in the end these groups are still created by the algorithm. So they might be they might be really helpful for you or there might not be. It depends a bit on not definitely not necessarily your luck, but basically the the quality of the data that you feed in, the aims that you have and, and all of these different dimensions. Now with K and N, it's a similar idea. You're still interested in different groups of customers, but this time you already have pre labelled some of them. So for example you have one group of customers and they are young people with a good income and you want to offer them a mortgage. And then you have one group of customers and they are a bit older. They have a high income and you would like to talk to them about retirement planning. So you have these different groups and you know they exist. You have maybe strategies for them. You already kind of have names for them. You have your basically have these customer segments that you know are in your database, and then you have new customers coming in and you don't necessarily know how you want to treat them yet. So you would then use K and N to check which of these other customers in your database are they closest to? Are they closest to the youngsters who would like their first mortgage, or are they closer to the the retirement focussed people a bit older, with a good income? Something like that. So you're basically looking for each of these new data points, new customers coming into your bank, to which segment would you assign them? And you do that in a supervised manner because you know which segments you would like to get out of that. So I hope this kind of makes makes a comparison between the two approaches a bit, a bit, a bit better to understand, a bit better to understand who. Okay, so I would take five more minutes just on this because I think it's really important to think about this last question. When do you think you would choose one or the other? I was already touching a bit on that just now with our idea of do we know what we want to get out of it? Or don't we know what we want to get out of it? So that was especially true if you think about something like k means in comparison to k and n. I mentioned that very similar methods. However, we were talking about clustering as well. When do you think hierarchical clustering might be the better choice over Canyon? And what was the main difference between hierarchical clustering and K-means for example? I mean, what what is the result of a clustering algorithm? If you want that on your data. I hear some some murmuring. You can't. You're brave. Shout it out. Is it a word? It's the word missing. The dent or gram. So you remember the tree structure thingy. So instead of getting one clustering result, you got this kind of nested system of multiple results. Now we were talking about how do we decide where to cut this tree. But you don't have to. You don't have to cut the tree. You can use the whole structure as it is and use the dendrogram, the tree itself to explain the structure in your data. And that is one of the main differences between hierarchical clustering, partitioning, clustering, and also one of the differences between hierarchical clustering and k and n k and n is a one off solution. It's cutting. It's dividing people into groups. And that's your solution. With hierarchical clustering you're not necessarily interested in just one solution. You're interested in solutions at multiple stages of your data. So that can be really useful if you try to communicate something about a structure within your data. Okay. So that was our discussion on clustering. And so we're done with all of that. Now the second part of this lecture we'll talk about feature selection via dimensionality dimensionality reduction. So that's a mouthful. And it's related to what we were talking about when we're talking about regression. So you'll remember back then we were talking about regularisation and this idea of how are we regularising regression models? And the idea was that we wanted to select just the most important features for our model. So instead of using all the variables that we have, we just wanted to use a few of them, most specifically the ones that best explain our relationship in the model. Obviously that has a lot of advantages. Improves interpretability because you don't have to explain every single of your 200 variables. You can just explain like ten of them. Accuracy can really be improved by that as well, because you're just not muddying the waters that much. So this is a bit counterintuitive because you might think, well, more parameters is always more accurate in the model, but as actually the. The advantage of that is, is decreasing. So it kind of goes up and at some point you're just not getting much more accuracy, even though you're increasing the number of your your parameters. In some cases, even the parameters interacting with each other in linear regression can muddy up the accuracy of your model. So I had models where removing parameters vastly improve the accuracy of the model, because suddenly one of the parameters which was working against the others was taken out of the equation. So sometimes it makes sense to take out variables of your regression, and it's easier to optimise them for obvious reasons. Less parameters, less computational strain. So we were talking about lasso regression which regression elastic net in that context, all of them very similar. All of them had the idea of kind of penalising the number of regressors that we have in our regression models and therefore try to reduce them, either take them out completely with lasso regression or reduce them and make them smaller with which regression and elastic network elevation. So we are now following a very similar logic. Yes. Using variables like gender or country is actually generating bias when we select those variables. Or is it supposed to be included as in, for example, if it's the example we said about giving someone about kids, including their gender, in a data analysis, is it creating a bias on who we are giving the mortgage to, or is it helping us to eliminate the. Even better. I think it's illegal. I'm pretty sure you can't include gender in that model. So this was let's say we're not talking about mortgage. Let's say we're talking about sending them emails about something else. I have no right, no idea about legislation. So to get to get to your question, it's actually an interesting one, including gender as a variable. And credit scoring, for example, is not allowed by the regulator. So you can't use gender as a variable in deciding whether to give someone, yeah, a credit card, for example, or not or a loan or not. And that has you might think that has advantages, but it was actually introduced years ago as far as I'm no. So storytelling as far as I know, a colleague of mine is much more knowledgeable about that. So if you're interested in the topic, look up the the publications by Professor Galina Andreeva. Here in the business school. She does research on credit scoring, and she's interested in gender as a variable in credit scoring. So very interesting research, if I remember correctly, from a talk by her a couple of months or years ago, including gender as a variable in credit scoring models improves the accuracy of the model, and it actually improves the situation specifically for women who you might think that was introduced to, to support better. So the idea was we are taking our gender as a variable because women might otherwise be disadvantaged in getting loans, as was the case in the past. For example, we're talking like 50s or 60s nowadays. It's actually if you're including gender as a variable, it's improving chances for women in getting loans because they tend to be better at repaying them. So theoretically it would be better to include gender verbal, but you can't for regulatory reasons. Yes. So the question when we talk about bias, the model it it's also the question biasing towards whom and what direction, what are the advantages and disadvantages and what are the regulatory frameworks that we're operating in. So I hope that helps. To answer your question. My my example from earlier was probably not very well thought of. So let's take h out of that. On the other hand, including age as a variable is another interesting one. So H can also actually bias the model. Because if you're thinking about, for example. People in the labour market who are a bit older often struggle finding new employment because they are older and people think they might be going into retirement soon. So some variables are difficult to include. Yes, that's the bottom line. Okay. Yes. So let's talk about dimensionality reduction. So this can be seen as an alternative approach to feature selection via regularisation, which we just talked about, but it is very different in its main setup. So instead of applying regularisation and trying to select predictors, we now actually try to combine predictors into combinations. So we are creating new predictors out of our existing ones by combining them specifically by creating linear combinations of them. And we're trying to do that in such a way that we explaining the largest amount of the variability in the data. Now, that sounds a bit odd, but it will get much clearer later when I explain in more detail. There are two things that we typically differentiate between factor analysis and principal component analysis. And they work very similarly from an underlying mathematical notion. So if you understand one you understand both. But they are used differently in context. Specifically, factor analysis is often used in, for example, the marketing literature and a lot of the social science literature and PCA principal component analysis is often used more in, for example, engineering, Stem, the sciences, and similar, but the underlying logic is the same. So let's talk briefly about factor analysis. And the idea here is that we're trying to reduce the number of predictors or the dimension of the data, because we think there is some kind of underlying factor driving multiple of those variables. So we think there's an underlying unobserved or latent factor which through which all of these variables can be explained. We often get this problem in surveys. So if you ever filled out a survey, you'll notice that a lot of these questions sound oddly similar to each other. So they might ask about what you think is the same concept or the same thing with different words multiple times, sometimes even with kind of switched around scales. We do all of that on purpose in marketing. Some of these reasons are one. We think they are the same concept. That's factor analysis, but we're not entirely sure. Or we think they are the same concept, but we don't know whether you also think the same. And we also try to measure the same thing multiple times throughout the survey, because you might get bored and kind of just start taking the middle box. So we want to kind of check your attention. So that's also why we kind of switch the scales the other way round, to see whether you're still paying attention or whether you just start taking all the four or the three or whatever on your scale. So the idea here is we think there's a concept. We're trying to measure the concept through multiple different variables, for example, multiple questions in a survey. And this belief system is really important factor analysis. And it's kind of the main difference between factor analysis of PCA and PCA. We don't care. We don't care whether there's an underlying factor which is described through the same variables. We just want to know whether we can combine them into one, because they measure kind of the same thing in the data, the same kind of variability in the data. That's all we care about. We don't care about the belief system. We don't care about the underlying factors. We're just trying to combine and reduce the dimensions. So that's why basically they are doing the same. Thing. But for different reasons. And that's why we sometimes differentiate between the two. So yes, they have a different reasoning work in the same way. We talk about PCA only here because factor analysis works in the same way. But we first have to do a little bit of linear algebra to understand what we're talking about. So we'll talk about two wonderful concepts today eigenvalues and eigenvectors. And this is finally something I can pronounce with absolute confidence. Because the word eigen in this is German. So eigen in German means approximately own or its own. So we would say this is basically the eigen value, its own value or the eigenvector, it's own vector. So it's finally a concept which I can actually pronounce similar to Gauss Gaussian. I used to say Gaussian because the name Gauss is also German. So there you go. Yes. So what actually is an eigen vector on an eigen eigen value? If we have some kind of square matrix A, and we multiply that by some vector x and this is equal to some scalar. So a single number times that same vector. Then we would call lambda here an eigenvalue of a and x and eigenvector. Now this does. This might seem a little unimpressive, but it's actually really interesting. It basically means if you multiply this matrix with some vector, it's the same as multiplying the same vector with a singular number instead of the whole matrix. So it's actually really interesting concept. So yes, we can find that by looking for the identity matrix, multiplying that, solving that not really a solution etcetera etcetera. Really interesting. If you want to actually calculate your eigenvalues and eigenvectors. So for example, we have a numeric example here which makes it a little easier. If we have this square matrix here and we're interested in calculating the eigenvalues of that, we can solve that as follows. We multiply it with its eigen, not with its eigen, with its identity matrix. There we go. Different concept. And we do solve that for lambda. I'm not going through the equation because I'm sure you can solve that yourself. You're basically multiplying the diagonal. You're summing it. Up. Oh. It's mine. No, mine. Yeah, yeah, yeah. She'll be right. Okay. Yes. So if we solve that, we do get our two eigenvalues, in this case three and two. And if we plug in these eigenvalues until we equation from earlier we can calculate our eigenvector. This is all very interesting for you. I'm sure. And I'm sure you've all overseen that. I put that here for completeness sake, because the calculation of that is. Not important, but I think is much more important is understanding the the. Next one. Okay. Yes. So let's skip over understanding what a covariance matrix is, which I'm sure you have. You've all seen it's basically a matrix of correlations and covariances. Yes linear combinations. We've seen that as well. So let's talk about PCA. So we've talked about PCA. Basically we want to seek simplification. We want to reduce the dimensionality of the data, and we want to cover what is happening in the data with less dimensions. So the idea behind that is it actually allows us to reorient the data in such a way that less dimensions are necessary to explain it. We'll also talk about the concept of projection in this in this topic. And I think this is basically at the heart of PCA. If we put that as an example. PCA allows us to aggregate. In your variables. The components. The compass or the comet. Sorry, let me start that sentence again, because it's I've written in a very odd way. PCA allows you to aggregate in new variables, the common information that are shared by the original variables. Basically, we're creating new variables, which we call our components, our principal components. But we want to still have the same amount of information that we had in the data earlier. So we are combining the original variables, and then we are ordering them in terms of the information that they actually convey. So in this case we can see here we have three original variables. And the different colours are basically the information that is hidden between that. So if we think about factor analysis that would be the different concepts that are captured for these different variables. So for example these three different questions and a questionnaire what kind of concepts do they measure. So these all three measure the same concepts to different amounts. And if we want to still be able to show the majority of the variation, the majority of what's happened in the information, in this data we first divide that into the three different components. We can see here 60% of the information that is captured through these variables as green, 30% of it is orange, and 10% is blue. So this is the same information. We're just taking that and kind of joining all the green parts, joining all the orange parts, joining all the blue parts. The idea behind PCA is now that instead of taking all of these components, why not leave out the last one? Because it's just 10%. We're not losing much if you're just leaving out that last component. We're basically leaving out 10% of the original information. But with reducing the dimensionality of the data by one suddenly would only have two dimensional data, two components. And this is the basic idea of PCA. We first have a look at what is the information available. How can we combine the information into linear combinations of it, so that they are kind of only measuring one direction of the data, one factor of the data. And then we can think about how much of this do we really need in our data. Is there a way of reducing that and therefore reducing the dimensions of our data? So why did we earlier cover the mathematical logic behind that? The reason for that is if we have a data space. We have n observations and we have some correlated vectors. So these are basically our variables here. So we have some kind of points base. Then we can calculate the covariance matrix. So we can look at how we lay to it. Are these different these different data points actually. And we can calculate that as our covariance matrix. And we're now looking via the principal components method of finding linear combinations. We call them z among these vectors in such a way that they are uncorrelated with each other. So this is the idea uncorrelated. So this green space has nothing to do with this orange space. Earlier they were all kind of mixed up and muddled up. And we can do that. By finding. The eigen vector of our covariance matrix. And with the note that as a. So these are. Then you. To create our principal components. This sounds very odd, and I hate explaining it via this concept because I never understood PCA when when I was taught the concept until actually someone just showed me a picture of it and it's really easy. Imagine we have a two dimensional space. Okay, so we have one variable XX1 with two second variable x two. And we have a point pattern here. So these are just the points that we collected on our data. And you can actually already see if I if I ask you is there a linear trend in this data. You would say yes obviously there is. It's like this direction right. It's going this pattern. So what if I told you that the principal component that we're looking for is just this line? It's exactly this line that we're going through it. Because what would a line through this pattern show us? The largest amount of the variance of the data. So we're interested in the spread of the data, the variance of it. How can we capture most of the variance of our data? It's in this direction. So we were drawing basically a new system, a new coordinate system through a data in such a way that we're capturing. The variance of it. And then we are drawing because we know our components have to be independent from each other. Your talk online through that. And then we have a new coordinate system going through the data. Now you ask me, okay, but how does this relate to dimensionality reduction? This is still two dimensional. Even if you take that space and kind of turn it a little bit, it's two. It's still two dimensional. But if you then think, okay, this would be the most variance that we capture, and this is the remaining variance that we capture. What if we just say, okay, let's let's forget about that. Let's forget about kind of the spread up and down. We are interested in the spread left to right. Because that is the really interesting part of the data. We don't really care about this kind of small spread around it. We care about this big spread. And that way you could reduce the dimensionality of the data by just looking at the spread along one of the axis. And this is really the idea about PCA. So you're looking at the largest spread, the largest variance in the data. And you find that via the eigenvector of the data, because the eigenvector of the covariance matrix. And if you remember, the covariance matrix is nothing else than the description of your data spread. So I thought, I thought that a really much easier idea behind that, that you think about how is the data spread? How can we best describe the data spread through a new coordinate system? Yes. So if you think about the eigenvector spacing, the describing the direction of the spread. So that would be this line here. And then the eigenvalue of that is the strength of that spread. How far does the data spread along that. And therefore, if we wonder how many PCAs or how many PCs principal components we actually need, we're looking at how big is the spread or how large is the eigenvalue of these eigenvectors. And we take those that have the largest ones. So the largest spread, the largest eigenvalues. We basically order the eigenvalues. We look for the largest ones typically like 70% maybe 80%. We say, okay, we don't care about the ten or the 20. We don't care about the small spread here. We care about the large one. So we reduce it to reduced dimension to capture just 7 to 80% of the spread of the data. Now in two dimensions, that kind of seems a bit easier, but it works the same way in three dimensions. So if you still think about. It's still all about projection into into lower dimensions. So when I say we forget about this axis here, we just care about this axis. Each of these points could be projected down on that axis. So we don't care about that. We say there's only one dimension. We smoosh it all down. So always think about the smooshing effect. We're basically smooshing the data down into just one component. And you can do the same from three into two components if you have two data points. We can even visualise that. Here are two data points. You can smoosh them down into two dimensions onto a flat surface. So the key thing I want you to remember is the idea is we're interested in how much of the variance are we able to explain with a smaller number of components than original variables? And I mentioned earlier, a rule of thumb is we would like to explain like 80% of the data ish. It can be 70, it can be 90. It depends on your application. But you're looking for how many components, how many dimensions, how many eigenvectors do I need to explain? 80% of the total variance of my data. And you can see that via the eigenvalue. So this would be one of these scree plots. That's an odd name, isn't it? Where you have the number of components on the x axis and your explained variance on the y axis, and you will recognise this plot because it's really similar to our elbow plot for K-means. So we are again looking for some kind of feature of some kind of parameter. It's not k in this case. So yay! We cheered, but it's still a number of components we're looking for, similar to K-means, similar to K. And and we're looking for explained variance in our data through that. So you'll see what that actually looks like in the computer lab tomorrow. So we'll go through that in the computer lab. You'll see how to actually create these plots. Then look for the number of components. But as long as you understand this logic, this idea of looking for how to explain most of the variance and we orientate the. The coordinate system. There we go. As long as you understand that I'm happy. Yes. Insurance components for. Yes. So let me think about that. Hmm. Should it be lost in variants? Let me, let me let me check that. That's very odd. Hmm. Let me think about that for five minutes. And I will come back to you. Yes. Okay, let me just finish that up because I think we're almost done. So an important point about PCA. It's not scale invariant. That makes sense. We just saw our coordinate system. We were interested in spread differences. So obviously it's not scale invariant. So you have to rescale your variables in order to be able to use it. Additionally, if there are large differences between the variances of the scales, then that will dominate which components you use. It also makes sense if you're interested in the spread and a lot of variables that show a lot of variance. Those will be kind of driving these early components. So always standardise your variables. So we also mentioned very, very early on, I think in the very first lecture when we were talking about supervised versus unsupervised learning, PCA is technically unsupervised because it doesn't take into account any dependent variable. So if you do PCA, you just take your X, basically your predictor variables, and you're trying to reduce the number of those or the dimensions of that matrix, and you're not necessarily creating components that are very good for prediction, because the model does not look at the actual predicted value or class for that. And a big one, these components might not be very interpretable. So we were talking about this earlier when we were looking at this plot. This component one. It's really nice. It explains 60% of the variability. But what is it? It's like like part of that I guess and parts of this. But if these are two questions in a questionnaire, for example, you don't really know what that is. You can try to explain it, you can try to interpret it. But in the end components, principal components cannot be interpreted with certainty. You can talk about loadings. So that would be the loadings of the variable for each of these components. How much impact does each of these variables have on each component. And try to interpret it that way. But in the end it's really difficult to interpret. And I also think this is one of the reasons why factor analysis is kind of has been developed not away from PCA, but has been developed as a discussion point at all because it's trying attempting to interpret the idea of these factors, which in this case would be our components, by thinking about underlying concepts, underlying causes driving them. But in the end, you're not sure. You can't really interpret them as well as you could. You could the war variables. You have the war variables. It's very easy to interpret PCA. Less dimensions, less interpretability. Let's talk briefly about principal component regression. It's a really simple concept. So we'll breeze through that because the idea is just that instead of using all of your predictors, your first using PCA to reduce the number of possible predictors and then build a regression model using your components instead of the variables. So instead of basically using your 100 variables, for example, you use PCA. You see 20 components are completely fine. So these are just combinations of all of these variables that we had earlier. And then you have a regression model with just 20 factors which is easier to train and easier to to build. And because we say this small number of components can explain 80% of the data, then you still have a really good accuracy of your model. Obviously, we said one of the most the best advantages of linear regression is interpretability. So you are losing quite a bit of that in that. But in many cases it can be really advantageous if you think, for example, you know, what these principal components mean or interpretation is just not that much of a concern for you. You just more interested in accuracy and especially computational efficiency. So yes, if our assumption underlying holds, then fitting at least square models will actually lead to better results. If we fit it to the components, that will be our Zs instead of our X, because most or all of the information in our data is in our components. That's how we create PCA. 80% ish of a variance should be in the components. So this can be really good to mitigate overfitting for.