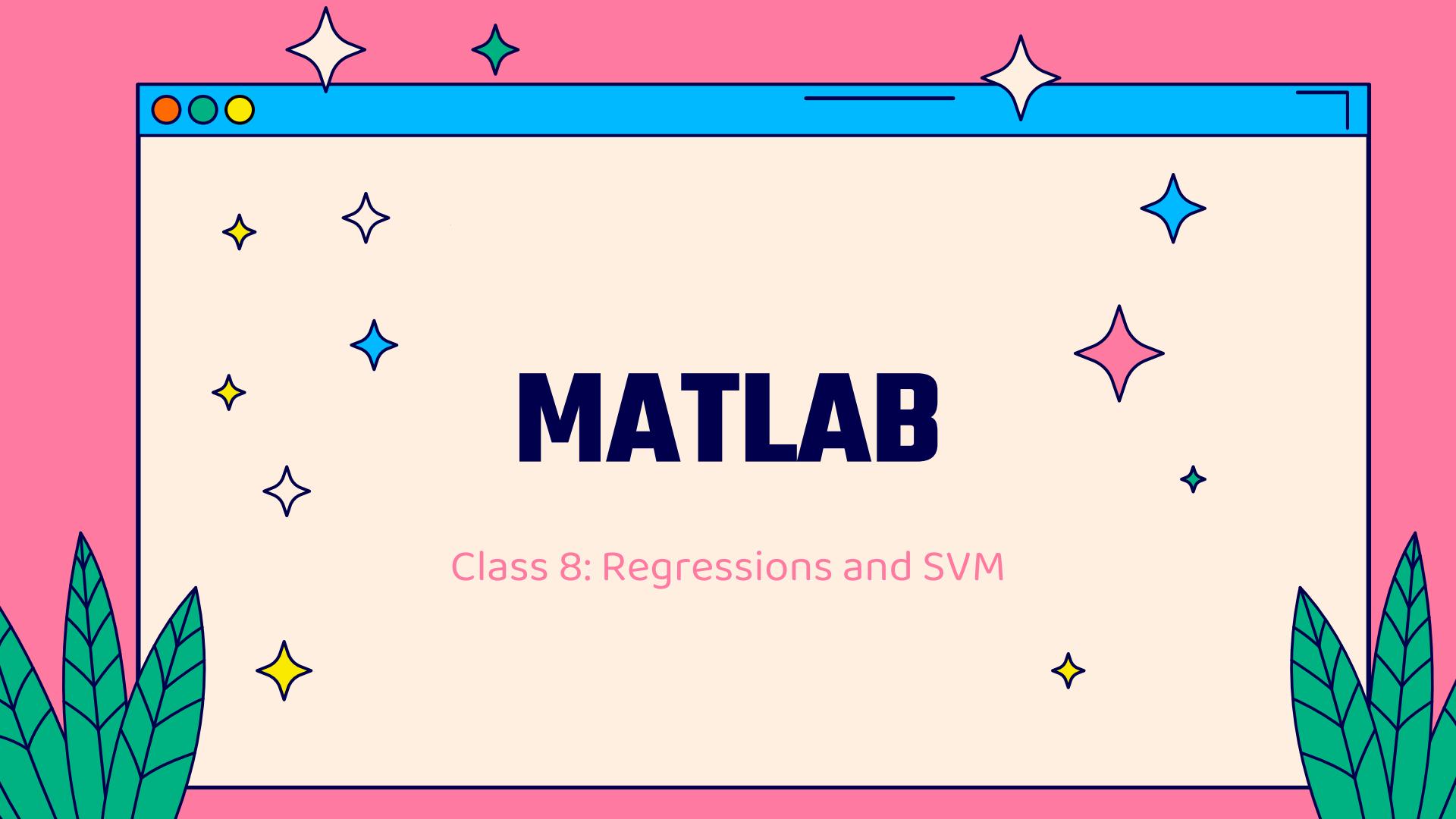




MATLAB

Class 8: Regressions and SVM





What is one of the ultimate goals of a model?

PREDICT



**Computational models make predictions based
on hypotheses.**

They predict the state of an outcome variable (y) given new
information x or at a new time point (x_t)



How do computers learn?

In machine learning and computational modelling, the way you teach your algorithm to predict can take one of two forms: **supervised** learning and **Unsupervised** learning



How do computers learn?

supervised learning takes place when computers/

algorithms are fed examples of data to learn from

Much like when you were a little kid, you were taught the

alphabet with examples of words that start with each

letter

Examples include regressions and SVM



How do computers learn?

Unsupervised learning takes place when computers/
algorithms learn without examples, but through trial and
error without the need of labels

Much like when you learned how to ride a bike

Examples include k-means (see class on clustering)



—

What is a regression ?

To put it simply a regression is a set of statistical tools used to describe and quantify the relationship between dependent and independent variables

Dependent variables (y)
are the outcomes we wish to predict based on
independent variables

Independent variables (x)
are the covariates/ predictors / features we
think change the value of the outcome variable
in a quantifiable way



Types of regression models

Linear Model:

Used when the outcome variables are continuous

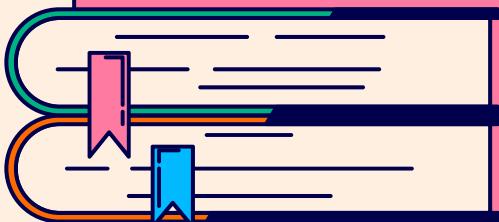
Logistic Model:

Used when the outcome variables are binary

Hierarchical Model:

Used when your data has a defined structure (i.e., variables are nested)

*** not mutually exclusive



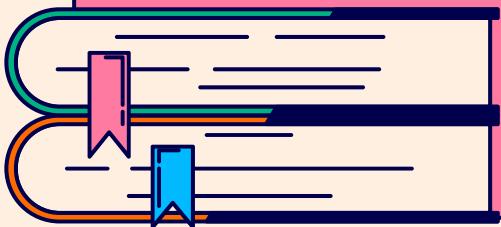


Linear regression models

Intercept → where the y-value of the regression line for an x of 0

Slope → how steep the line is (for every 1 unit increase in x, how much do you expect y to increase)

$$y = \beta_0 + \beta_1 * Age + Residual$$





Linear Regression Models

Coding of your ordinal predictors is important for the interpretation of your betas:

Linear coding: 1 2 3 assumes that there is a linear relationship between the orders (i.e., $x=1$ is half of $x=2$, etc.)



Linear Regression Models

Coding of your categorial predictors is very important for the interpretation of your betas:

Dummy coding

Effect Coding

Dummy Coding

Attention (X1)	Difficulty (X2)	Var Attention	Var Difficulty
High	easy	1	0
Low	easy	0	0
High	easy	1	0
High	hard	0	1
Low	hard	1	1
High	hard	0	1

- Let's say we have two categorical variables:
 - Attention (high vs low)
 - Difficulty (easy vs hard)
- Dummy code them as 0 and 1



Dummy Coding

Attention (X1)	Difficulty (X2)	Var Attention	Var Difficulty
High	easy	1	0
Low	easy	0	0
High	easy	1	0
High	hard	0	1
Low	hard	1	1
High	hard	0	1

- The intercept β_0 is the mean of y for all conditions when they are 0 (i.e., Low--easy)
- The slope β_1 is the difference between High and Low Attention for the easy task
- The slope β_2 is the difference between the easy and difficult task for low attention
- The slope β_3 is the interaction

$$y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$$



Dummy Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

$$y = \beta_0 + \beta_1 \mathbf{0} + \beta_2 \mathbf{0} + \beta_3 \mathbf{0} * \mathbf{0}$$
$$y = \beta_0$$

$$y = \beta_0 + \beta_1 \mathbf{1} + \beta_2 \mathbf{0} + \beta_3 \mathbf{1} * \mathbf{0}$$
$$y = \beta_0 + \beta_1$$

$$y = \beta_0 + \beta_1 \mathbf{0} + \beta_2 \mathbf{1} + \beta_3 \mathbf{0} * \mathbf{1}$$
$$y = \beta_0 + \beta_2$$

$$y = \beta_0 + \beta_1 \mathbf{1} + \beta_2 \mathbf{1} + \beta_3 \mathbf{1} * \mathbf{1}$$
$$y = \beta_0 + \beta_1 + \beta_2 + \beta_3$$



Dummy Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

$$X(1,0) - X(0,0) = \beta_0 + \beta_1 1 - \beta_0$$

$$X(1,0) - X(0,0) = \beta_1$$

$$X(0,1) - X(0,0) = \beta_0 + \beta_2 1 - \beta_0$$

$$X(0,1) - X(0,0) = \beta_2$$

$$(X(1,1) - X(0,1)) - (X(1,0) - X(0,0))$$

$$= (\beta_0 + \beta_1 1 + \beta_2 1 + \beta_3 1 - \beta_0 - \beta_2 1) - (\beta_0 + \beta_1 1 - \beta_0)$$

$$= (\beta_1 + \beta_3) - (\beta_1)$$

$$= \beta_3$$



Dummy Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

→ $X(1,0) - X(0,0) = \beta_0 + \beta_1 1 - \beta_0$
 $X(1,0) - X(0,0) = \beta_1$

→ $X(0,1) - X(0,0) = \beta_0 + \beta_2 1 - \beta_0$
 $X(0,1) - X(0,0) = \beta_2$

→ $(X(1,1) - X(0,1)) - (X(1,0) - X(0,0))$
 $= (\beta_0 + \beta_1 1 + \beta_2 1 + \beta_3 1 - \beta_0 - \beta_2 1) - (\beta_0 + \beta_1 1 - \beta_0)$
 $= (\beta_1 + \beta_3) - (\beta_1)$
 $= \beta_3$



Effect Coding

Attention (X1)	Difficulty (X2)	Var Attention	Var Difficulty
High	easy	1	-1
Low	easy	-1	-1
High	easy	1	-1
High	hard	-1	1
Low	hard	1	1
High	hard	-1	1

- In effect coding all columns of variables sum to 0
- To do this let's try replacing the 0's with -1



Effect Coding

Attention (X1)	Difficulty (X2)	Var Attention	Var Difficulty
High	easy	1	-1
Low	easy	-1	-1
High	easy	1	-1
High	hard	-1	1
Low	hard	1	1
High	hard	-1	1

- The intercept β_0 is now the GRAND Mean
- The slope β_1 is the difference between High and Low Attention across both difficulties
- The slope β_2 is the difference between difficult and easy across Attention
- The slope β_3 is the interaction

$$y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$$



Effect Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

$$y = \beta_0 + \beta_1(-1) + \beta_2(-1) + \beta_3(-1) * (-1)$$
$$y = \beta_0 - \beta_1 - \beta_2 + \beta_3$$

$$y = \beta_0 + \beta_1(1) + \beta_2(-1) + \beta_3(1) * (-1)$$
$$y = \beta_0 + \beta_1 - \beta_2 - \beta_3$$

$$y = \beta_0 + \beta_1(-1) + \beta_2(1) + \beta_3(-1) * (-1)$$
$$y = \beta_0 - \beta_1 + \beta_2 - \beta_3$$

$$y = \beta_0 + \beta_1(1) + \beta_2(1) + \beta_3(1) * (1)$$
$$y = \beta_0 + \beta_1 + \beta_2 + \beta_3$$



Effect Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

$$\begin{aligned}X(1,0) - X(0,0) &= \beta_0 + \beta_1 - \beta_2 - \beta_3 - (\beta_0 - \beta_1 - \beta_2 + \beta_3) \\&= \beta_0 + \beta_1 - \beta_2 - \beta_3 - \beta_0 + \beta_1 + \beta_2 - \beta_3 \\&= +\beta_1 + \beta_1 \\&= 2\beta_1\end{aligned}$$

$$\frac{1}{2} * (X(1,0) - X(0,0)) = \beta_1$$

Thus your main effect is twice as big, and your beta is half of the estimated effect



Effect Coding

Attention (X1)	Difficulty (X2)	Var Attention	Var Difficulty
High	easy	0.5	-0.5
Low	easy	-0.5	-0.5
High	easy	0.5	-0.5
High	hard	-0.5	0.5
Low	hard	0.5	0.5
High	hard	-0.5	0.5

- What about replacing 0's with -0.5 and 1's with 0.5



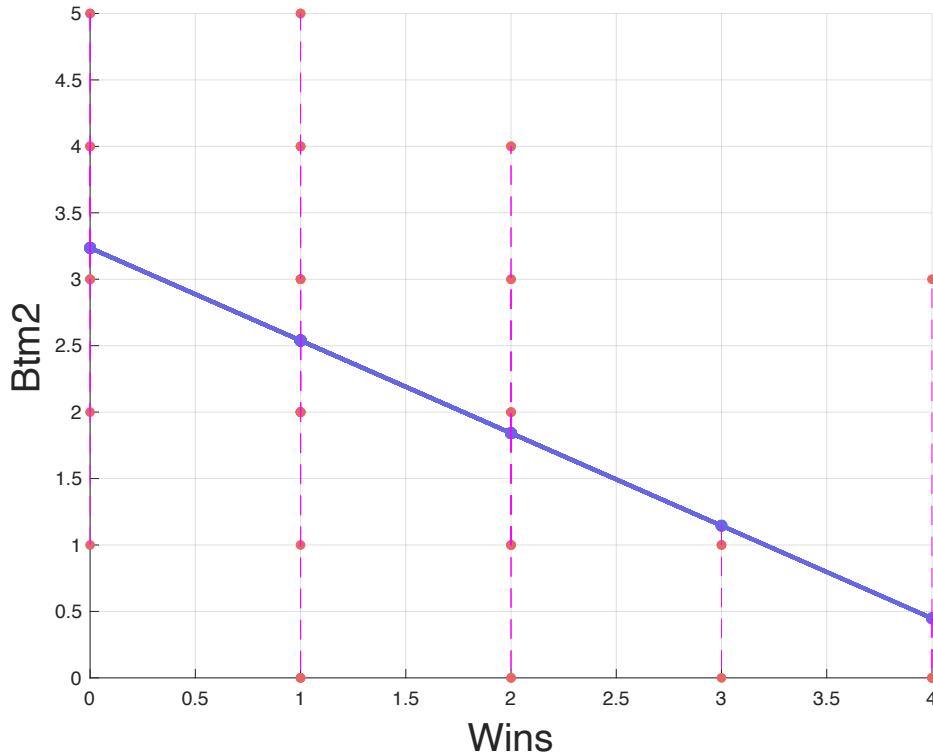
Effect Coding

Our regression model: $y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \beta_3 X1 * X2$

$$\begin{aligned} X(1,0) - X(0,0) &= \beta_0 + \frac{1}{2}\beta_1 - \frac{1}{2}\beta_2 - \frac{1}{2}\beta_3 - \left(\beta_0 - \frac{1}{2}\beta_1 - \frac{1}{2}\beta_2 + \frac{1}{2}\beta_3\right) \\ &= \beta_0 + \frac{1}{2}\beta_1 - \frac{1}{2}\beta_2 - \frac{1}{2}\beta_3 - \beta_0 + \frac{1}{2}\beta_1 + \frac{1}{2}\beta_2 - \frac{1}{2}\beta_3 \\ &= +\frac{1}{2}\beta_1 + \frac{1}{2}\beta_1 \\ &= \beta_1 \end{aligned}$$

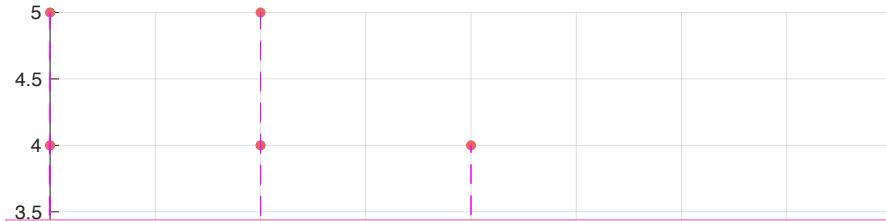


Linear Regression Models

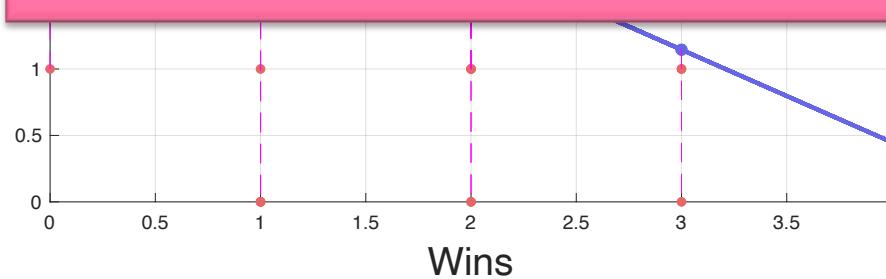




Linear Regression Models



You predicted y
what now?





Linear Regression Models

Regressions can be used in different fashions: testing statistical relationships, predicting future data, removing effects out of your data

After fitting regressions, you can compute and utilizes the residuals (i.e., the unexplained variance or y) for further analysis



Linear Regression Models

Because of how linear regressions compute effects (holding all other effects at zero), you can add nuisance variables into your regression to compute your desired effect while controlling for the effect of another uninteresting variable

$$\text{Alpha power} = \beta_0 + \beta_1 \text{Age} + \beta_2 \text{Attention} + \beta_3 \text{Sleep Quality}$$



Linear Regression Models

Because of how linear regressions compute effects (holding all other effects at zero), you can add nuisance variables into your regression to compute your desired effect while controlling for the effect of another uninteresting variable

$$\text{Alpha power} = \beta_0 + \beta_1 \text{Age} + \beta_2 \text{Attention} + \beta_3 \text{Sleep Quality}$$



controlling for alpha age effects / confounds



Multicollinearity

Is a problem when two or more of your predictors / regressors/
measures are related to one another (i.e., correlated)

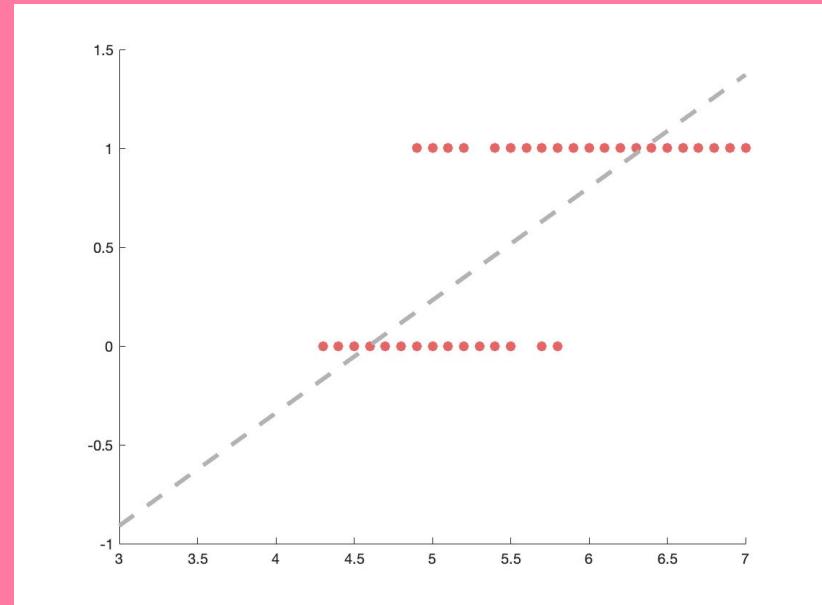
For example, participant height and weight

This is a problem because one of the assumptions of regressions
are that the measures are independent, and if this violated the
beta coefficients estimated become unstable and sensitive to
small changes



Logistic Regression

Linear regression work very well on continuous outcomes, but what happens when your outcome is categorical or binary? Fitting a line just doesn't cut it

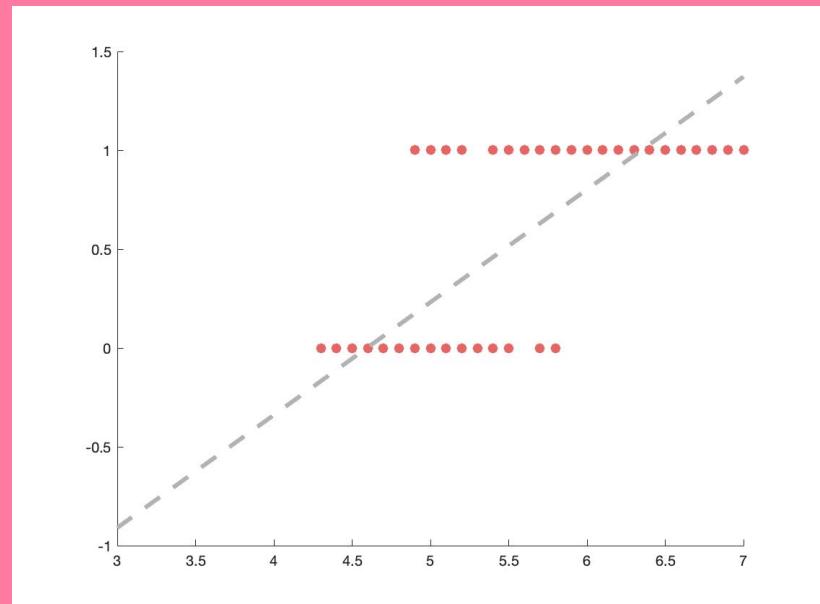




Logistic Regression

But if we fit a different
function to the data, it
might fit it much
better...

Here comes in Logistic
Functions (Sigmoid
Function)

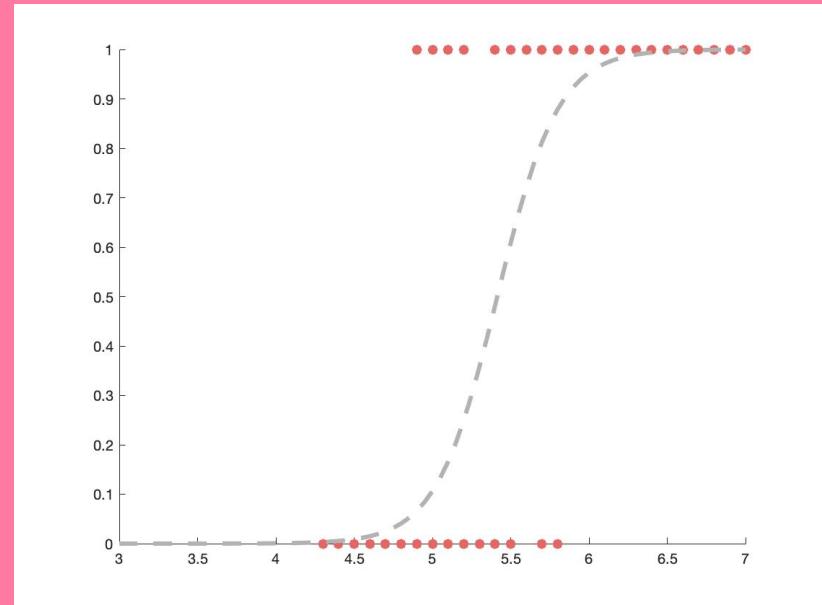




Logistic Regression

But if we fit a different
function to the data, it
might fit it much
better...

Here comes in Logistic
Functions (Sigmoid
Function)





Logistic Regression

Logistic regression run on the odds $\frac{p}{1-p}$ (specifically the log odds)

an Odds of 1 means equally likely to happen or not happen

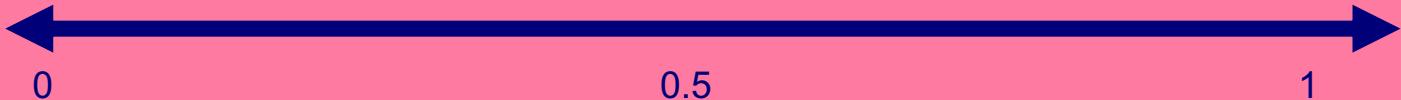
An Odds < 1 means less likely to happen

An Odds >1 means more likely to happen

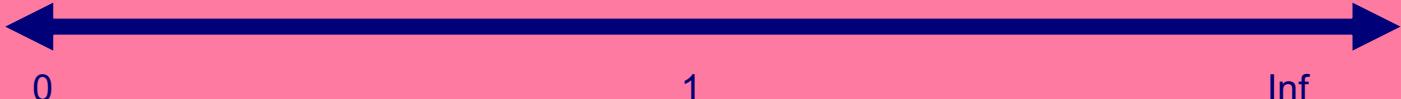


Odds

Probability

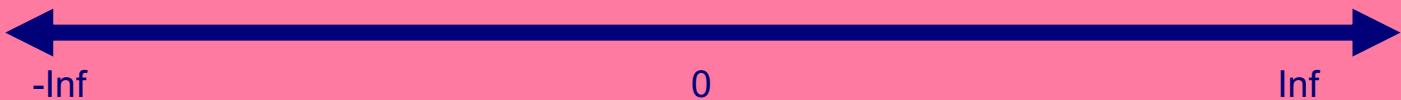


Odds



$$p/(1-p)$$

Log Odds



$$\ln(x)$$



Logistic Regression

Logistic regression are harder to converge and require more data to fit

$$\log_e \frac{p}{1-p} = \beta_{0j} + \beta_{1j} * Age$$

The Intercept is the log Odds of your event when x is 0
If the intercept is ZERO → you have a 50/50 chance
If the intercept is negative → you're less likely
If the intercept is positive → you're more likely



Logistic Regression

Logistic regression are harder to converge and require more data to fit

$$\log_e \frac{p}{1-p} = \beta_{0j} + \beta_{1j} * Age$$

The Slope controls how steep the relationship is
If the Slope is small → the relationship between x & y is noisy
If the Slope is negative → small values of X are more likely
If the Slope is positive → large values of X are more likely



Nested data

What is the best song to dance to?





Nested data

What is the best song to dance to?





Nested data

What is the best song to dance to?

Le Club

Muzique

Le Rouge

GAY

Club MED

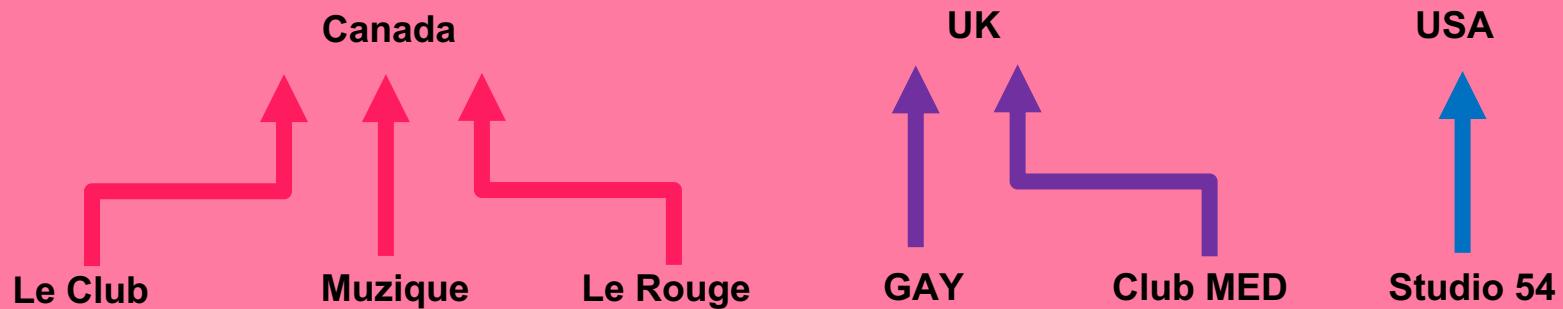
Studio 54





Nested data

What is the best song to dance to?





Nested data

Observations within the same country, for example, are similar to themselves than to a different country

When data is nested, it is important that your model takes into account the correlational structure of your data



Nested data

Regressions assume that all your predictors (i.e., measures) are **independent**. But this may not always be the case, specifically for **repeated measure** designs where participants see multiple experimental conditions



Nested data

The independence of each observation is often violated
in real life—data is nested in a big wonderful world

People are more similar to themselves, their family, their
friends, people from their city, background, country
etc



Nested data

Regression models need to consider the correlated structure of your observations to better deal with noise and explain your data better

Adding this info to your regression will allow you to better predict out of sample



Hierarchical regressions

Hierarchical regressions allow you to take into account the structure of your data to help better estimate your effects and deal with noise that can be accounted for by your nested data



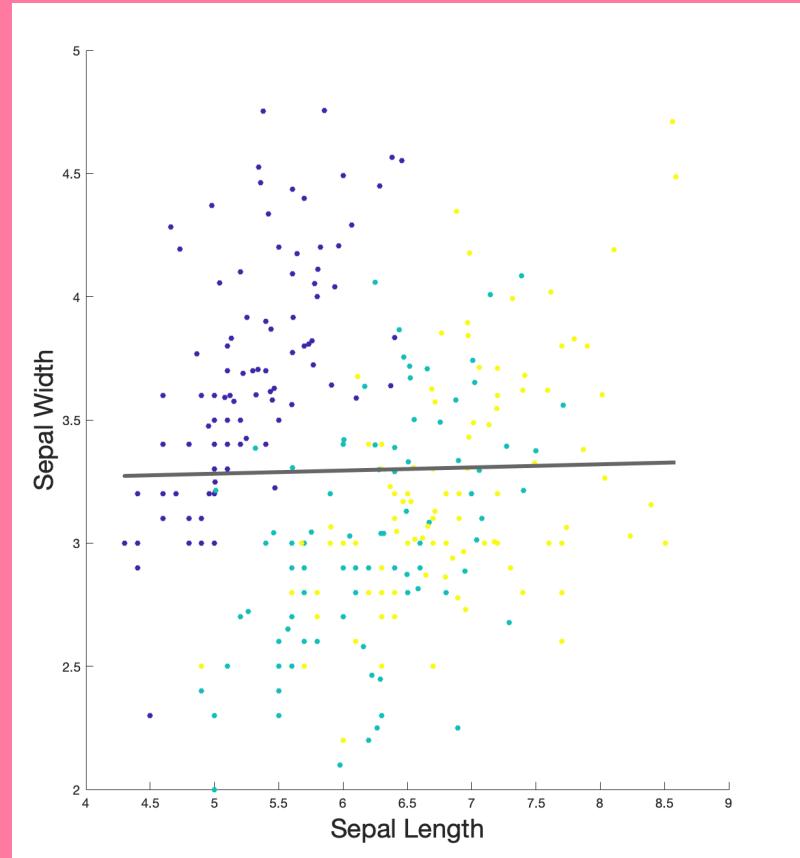
Hierarchical regressions

This is achieved by allowing Intercepts or slopes vary as a function of the levels/ nests within your data

For example, allowing every person in your sample to get a different intercept accounts for the fact that people have differences in response times, but fixing the slope of the regression lets you test the effect across your conditions

Hierarchical regressions

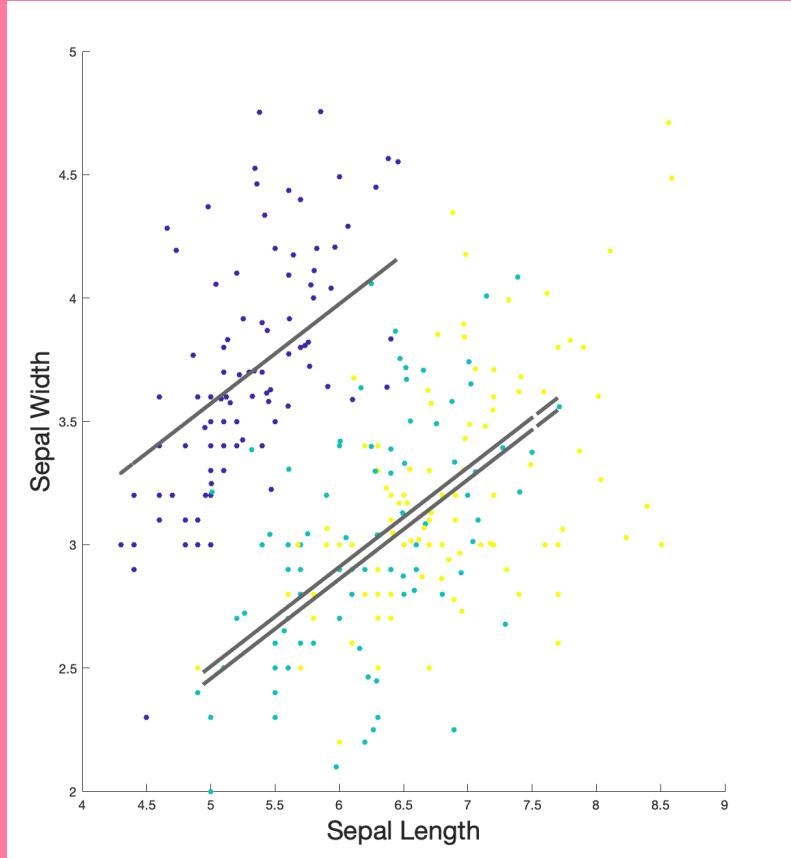
Fitting a line to our data sometimes does not capture the complicated effect we may have





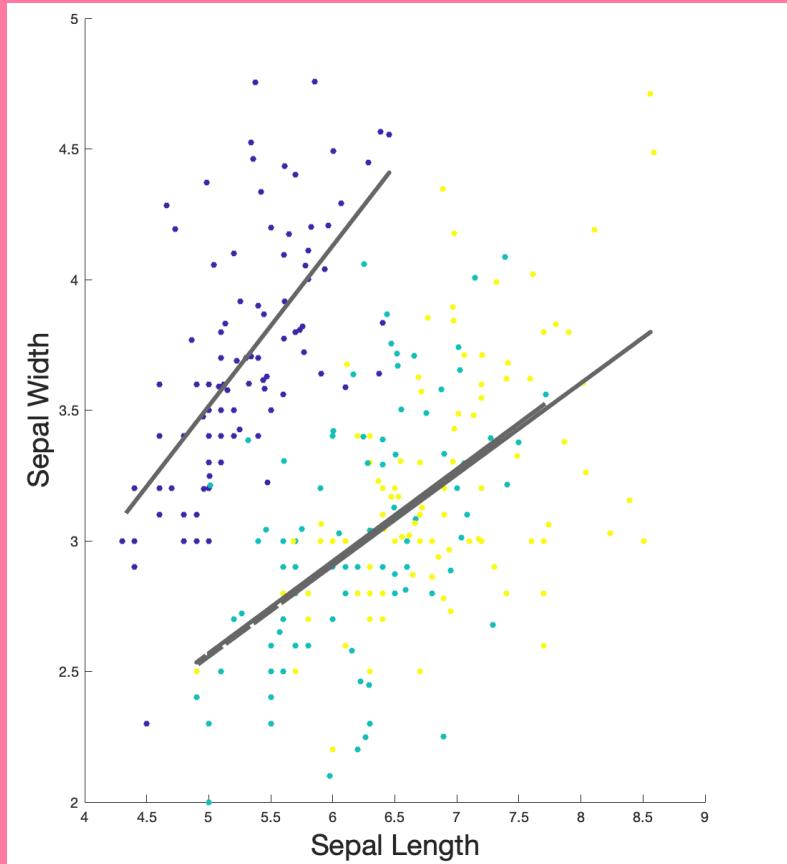
Hierarchical regressions

Random Intercept
Sometime our effect is
constant across
individuals, but they
may vary in their
baseline ability



Hierarchical regressions

Random Slope
Sometime our effect
changes across
groups too

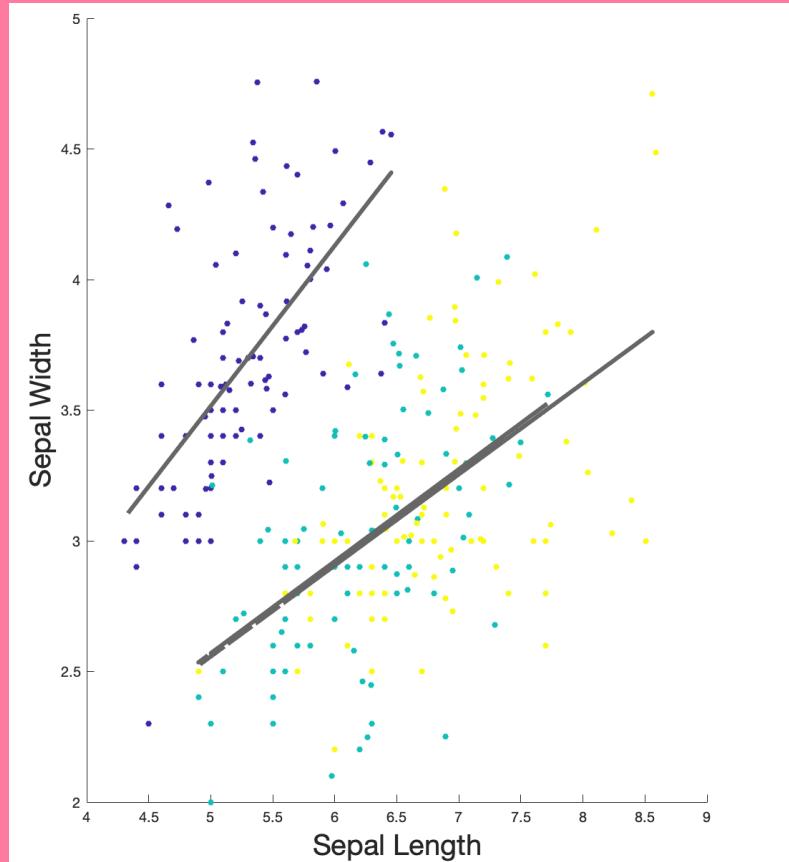




Hierarchical regressions

See

<http://mfviz.com/hierarchical-models/> for
interactive example
demonstrating fixed vs
random slopes and
intercepts





Fixed vs Random effects

Note that hierarchical models have both ***fixed*** and ***random*** effects

Fixed effects—assume you have every desired level of variable (e.g., experimental condition)

Random effects—assume you do not have every possible level you wish to test (e.g., participant)



Fixed vs Random effects

Fixed effects—sometimes referred to as variables you MANIPULATE

Random effects—sometimes referred to as variables that are sources of variation in your data that you do not care about

Model selection revisited

We can try and quantify a measure that balances both the **fit** of the model as well as **parsimony** (i.e., the number of parameters)

Akaike information criterion (AIC).

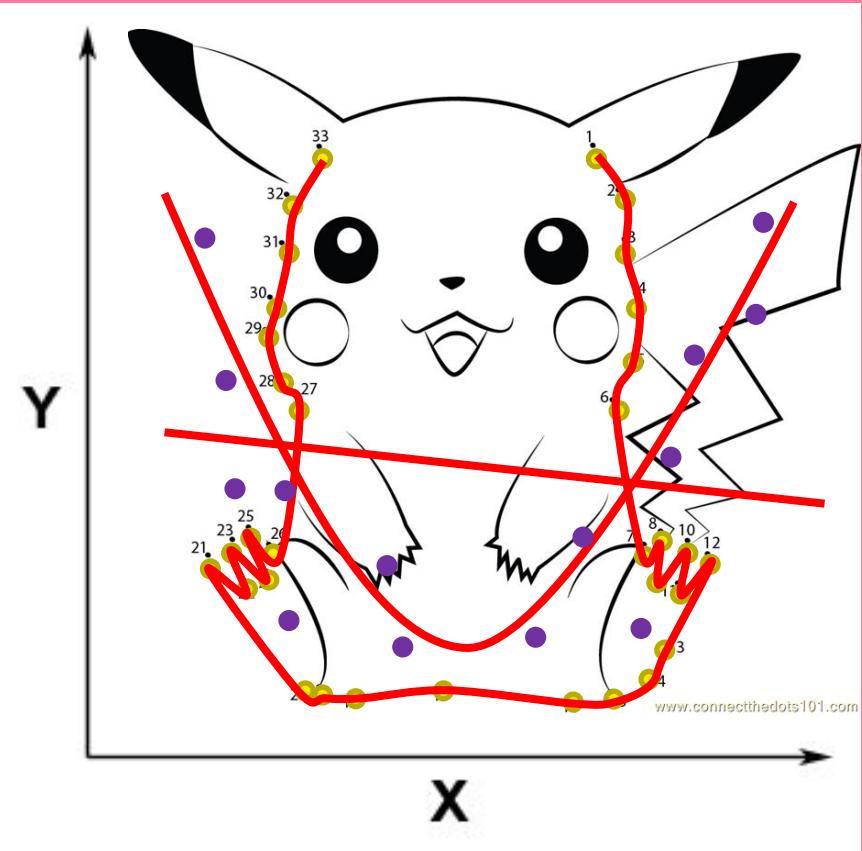
$$-2\log L(\theta) + 2k$$

Bayesian information criterion (BIC)

$$-2\log L(\theta) + k^* \log(T)$$

Where k is the number of parameters fit to T observations

Overfitting



$$Y = \beta X + \alpha$$
$$Y = \beta X^2 + \gamma X + \alpha$$
$$Y = \beta X^2 + \gamma X + \frac{\xi \sin(\tau(\sqrt{X} - \varphi))}{\sqrt[3]{e^{-\omega X}}} + \alpha$$



Support Vector Machines

Supervised machine learning algorithm that learns to classify two classes of data (can be generalized to multi class problems) given a bunch of examples of each class



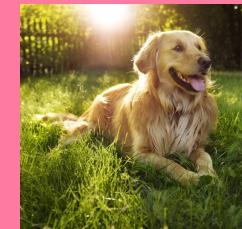
Support Vector Machines

It works using **support vectors** that help create a decision boundary (i.e., a hyperplane) along your multidimensional feature space to classify your inputs.

Support vectors are cases that are very close to the decision boundary (i.e., ambiguous cases)



Support Vector Machines



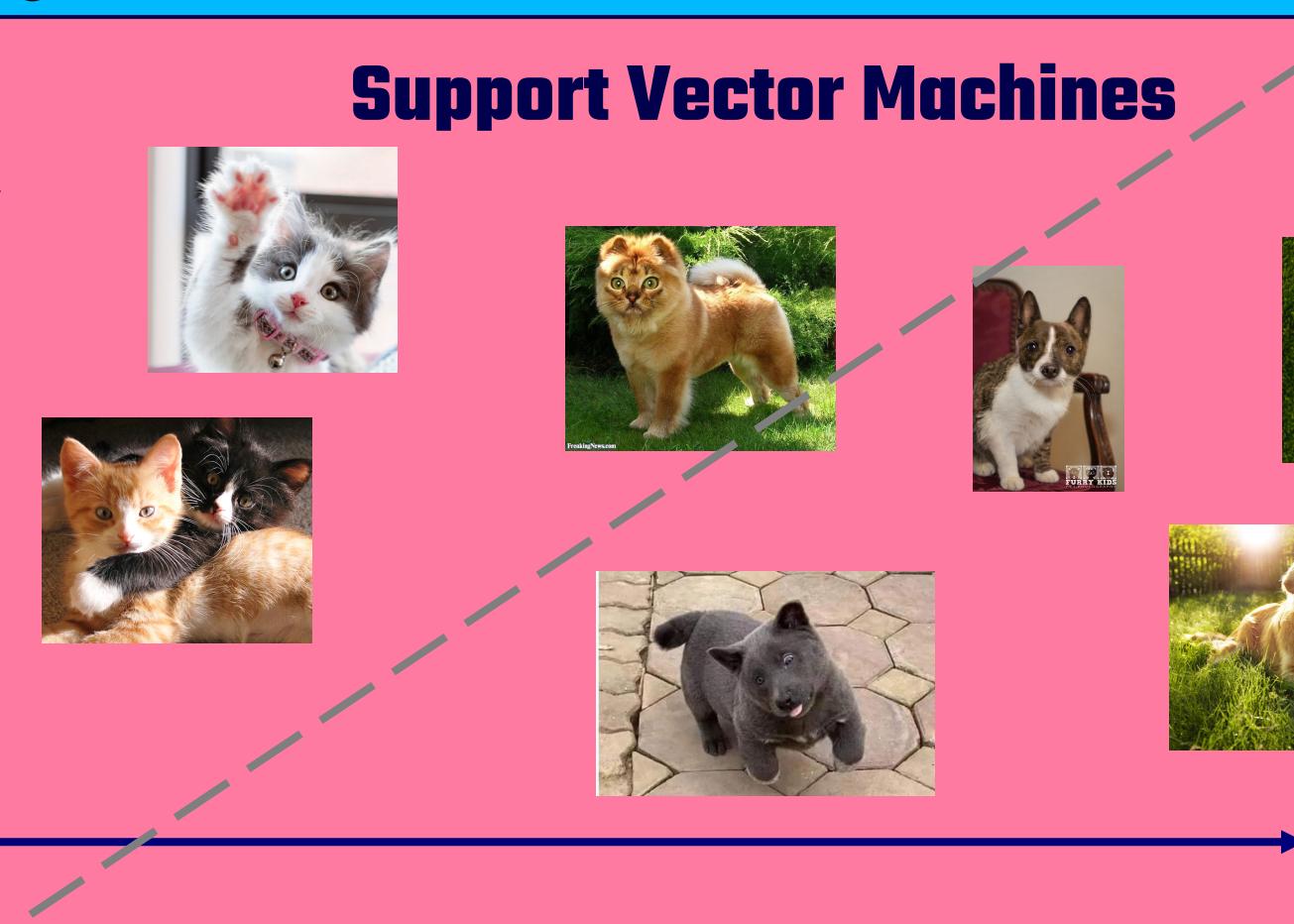


Support Vector Machines



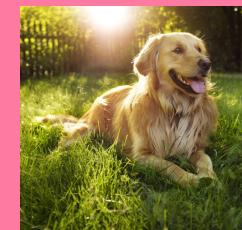


Support Vector Machines





Support Vector Machines





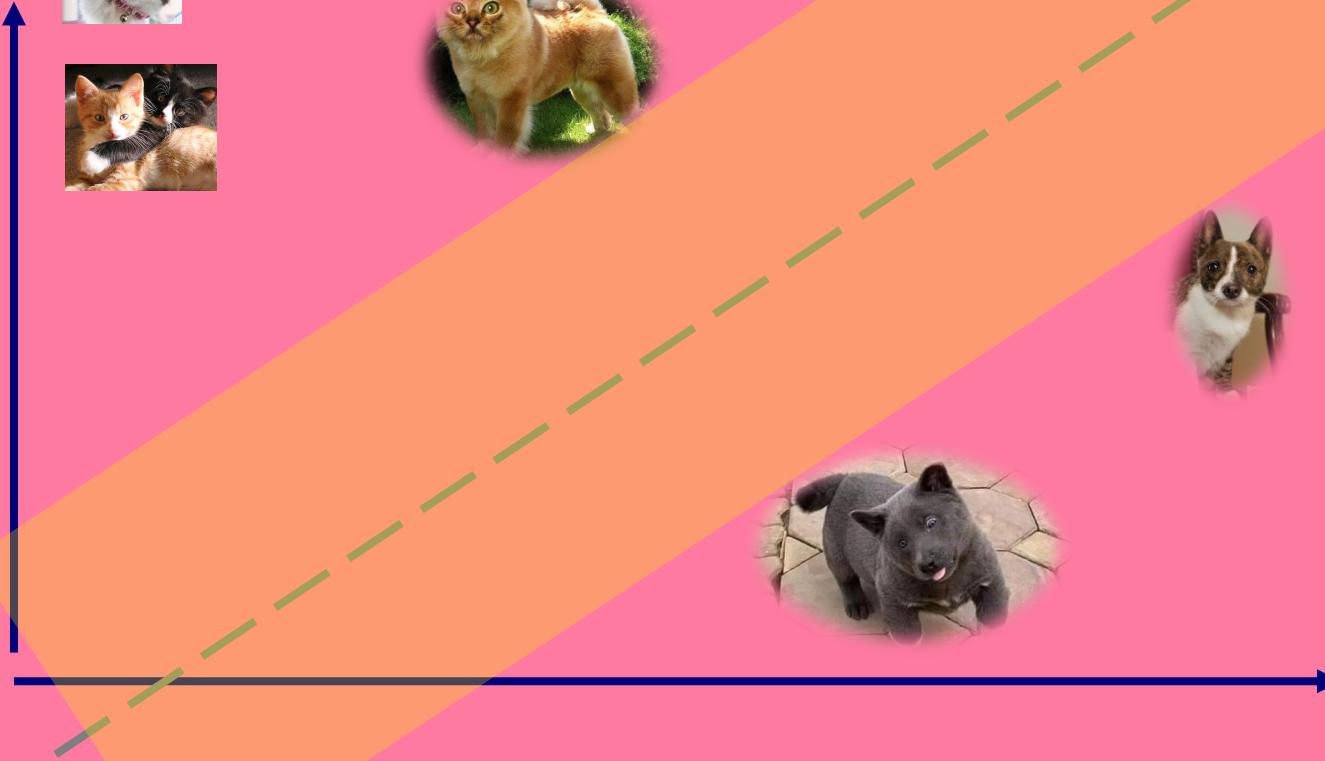
File Edit View

Support Vector Machines





Support Vector Machines





Support Vector Machines



SVM tries to find a hyperplane such that it leaves the widest lane between the edge examples

