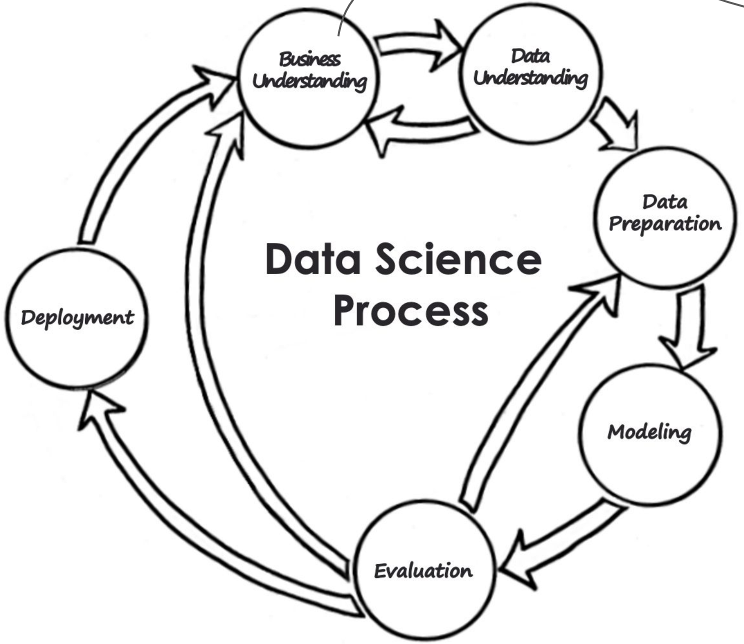
MIST 5400
Foundations of Artificial Intelligence in Business

- w2: The Data Science Process (technical tools)

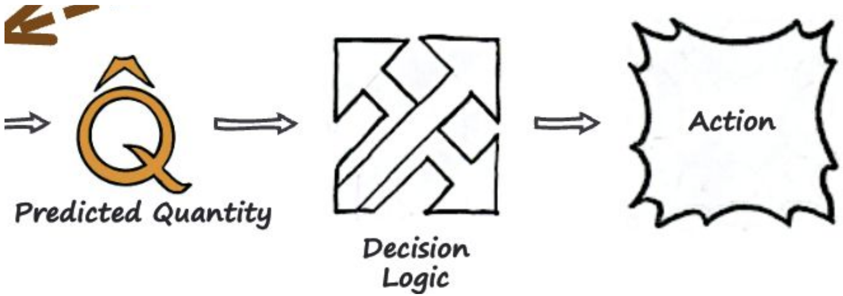
Pearl Yu

A Little Recap

- **Business Problem:** Customers are churning. We'd like to reduce it!



- **Business understanding:**



Action: If a customer, send promotional offer?
Decision logic considerations:

- Consider costs of offers
- People who're most likely to leave will leave no matter what
- Customers have different values. etc

Deriving the decision logic

- The expected value framework

We could predict this too, or let's assume it's the current plan price

Action:
If I have a customer:

Send offer E[profit| send offer] = Pr(stay| send offer)*(Value of customer - offer cost)
+Pr(churn|send offer)*(0 - offer cost)

= 1 - Pr(stay| send offer)

Let's assume it's decided already.

= 1 - Pr(churn| no offer)

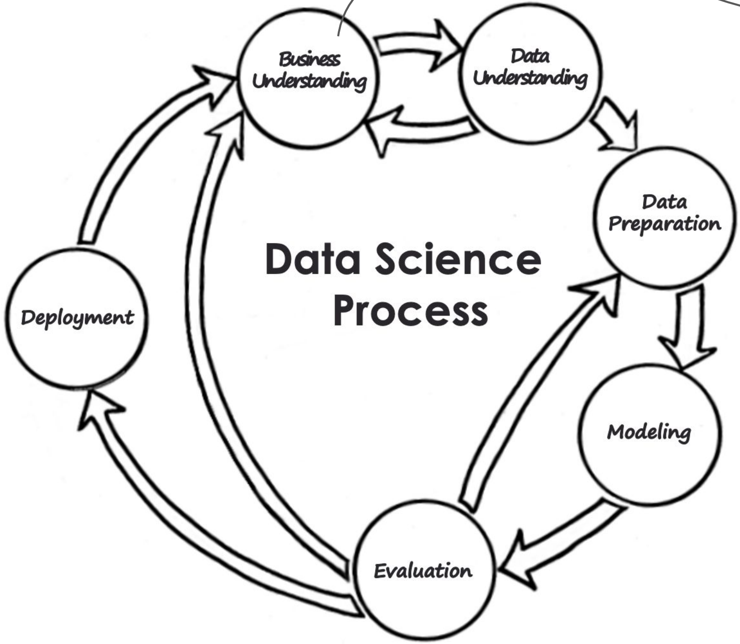
Don't send offer E[profit| Not send offer] = Pr(stay| no offer)*(Value of customer - 0)
+Pr(churn|no offer)*(0 - 0)

So, the unknown quantities to predict (target variables)?

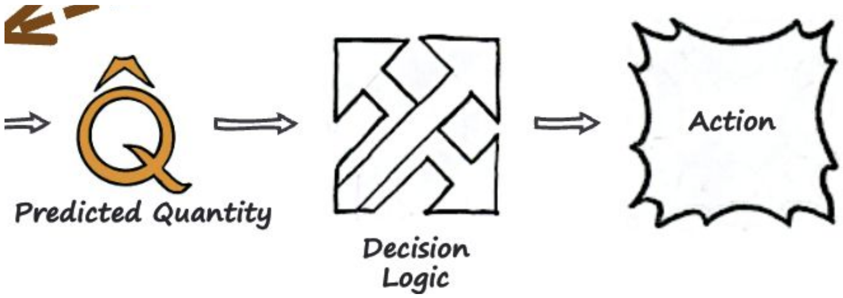
Decision logic:
E[profit | send offer] - E[profit |no offer] > a threshold, send offer.
How to decide the threshold: Could be 0, could be based on the budget limit, etc.

A Little Recap

- **Business Problem:** Customers are churning. We'd like to reduce it!



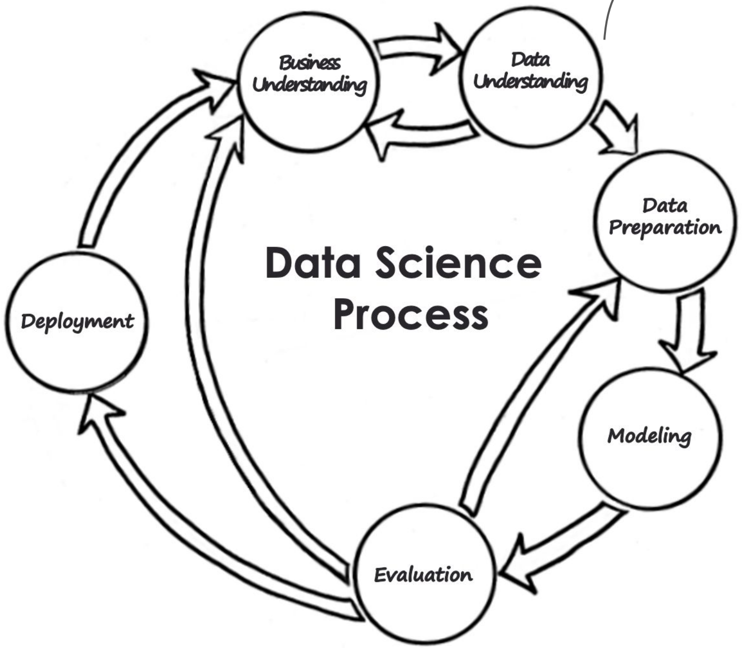
- **Business understanding:**



Action: If a customer, send promotional offer?
Decision logic
Target variables: Pr(churn|no offer), Pr(stay|send offer)

A Little Recap

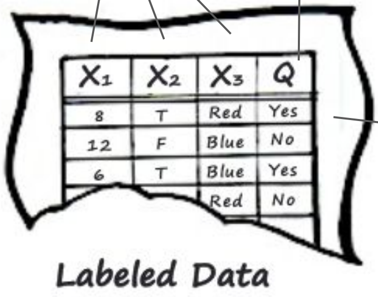
- **Business Problem:** Customers are churning. We'd like to reduce it!



- **Data:**

label/target variable

features/attributes



instance

- We need enough info that're predictive of the target variable.

Having the right data is important!

If I have a taco cart and past sales data of tacos. And I want to sell merchandise, like baseball caps. Can I predict the sales of baseball cap sales?

About having the right data

If I have a taco cart and past sales data of tacos. And I want to sell merchandise, like baseball caps. Can I predict the sales of baseball cap sales on a weekday?

Not really if you've never sold caps before, cuz you don't have any training data!

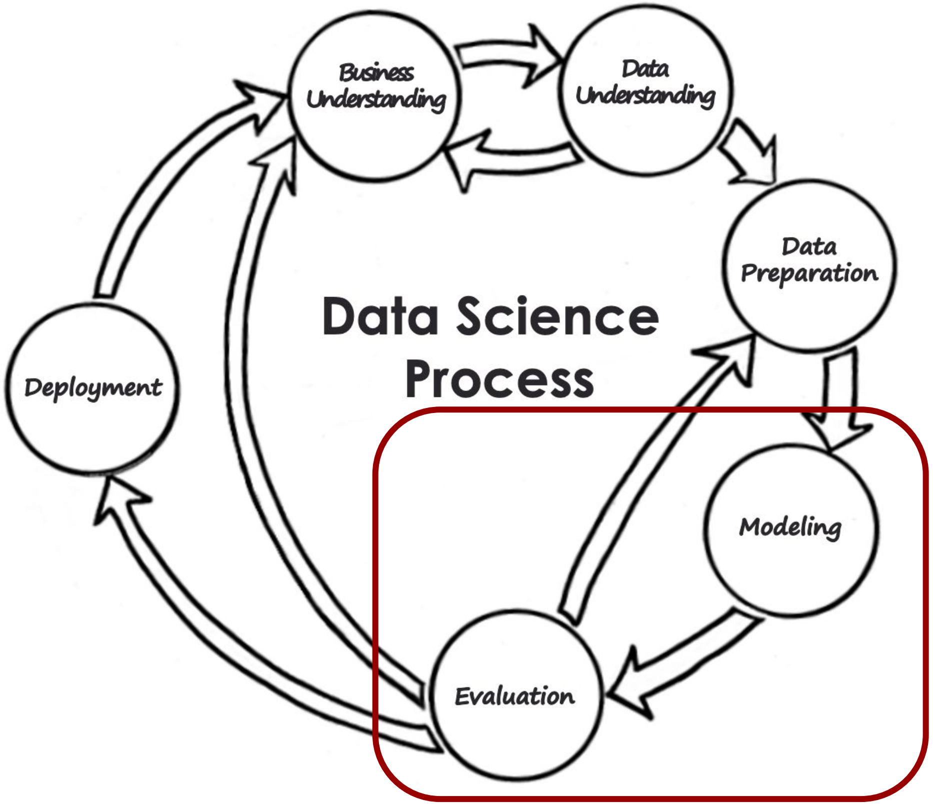
Could try sales of tacos as proxies, but won't perform that well.

Having the right data is important!

Pr(churn|no offer)

Pr(stay|send offer)
—> If you've sent similar offers before / randomly send offers to a small number of customers

The Data Science Process



Technical tools

- Supervised / Unsupervised
- The models (ML algorithms)
- The training
- The evaluation (Metrics / Overfitting)

Supervised v.s. Unsupervised Learning

Supervised v.s. Unsupervised Learning

Are LLMs supervised or unsupervised?

[The cat likes to sleep in the ____] → What **word** comes next?

We can create **vast amounts of sequences** for training a language model

● Context ● Next Word ● Ignored

[The cat likes to sleep in the]

[The cat likes to sleep in the]

[The cat likes to sleep in the]

[The cat likes to sleep in the]

[The cat likes to sleep in the]

Linear Regression

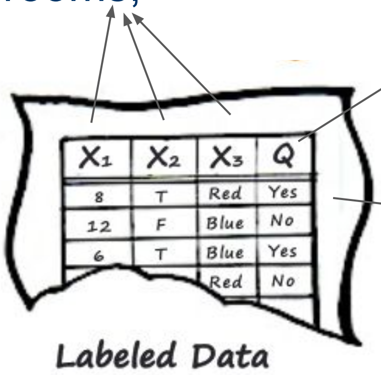
- Linear Regression: Predict a numeric variable from one or multiple other variables.
  - Simple Linear Regression
    - Only one explanatory variable (attribute)
  - Multiple Linear Regression
    - Multiple explanatory variables (attributes)

Linear Regression

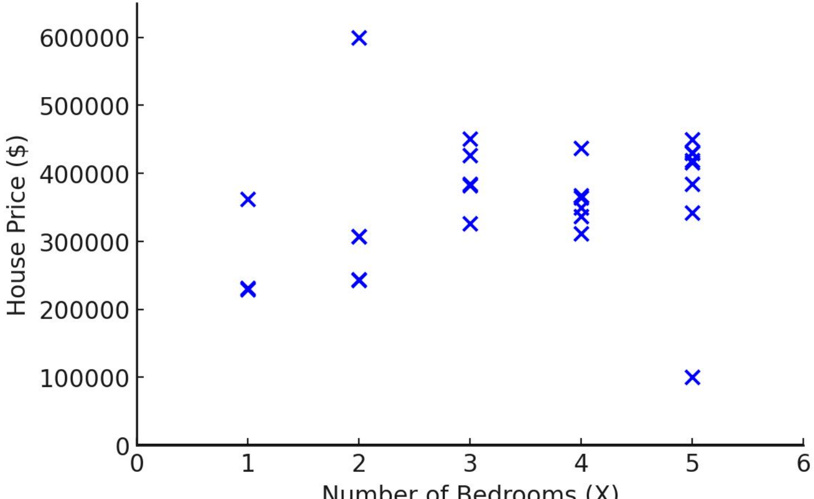
- Linear Regression: Predict a numeric variable from one or multiple other variables.
  - Simple Linear Regression
    - Only one explanatory variable (attribute)
  - Multiple Linear Regression
    - Multiple explanatory variables (attributes)

Num bedrooms, sqft

House price



a house



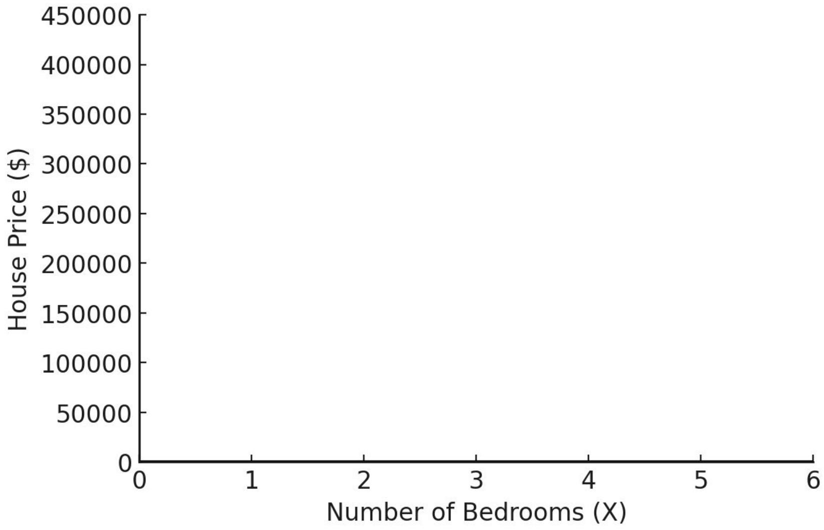
Linear Regression

- Any line is mathematically expressed as an equation with a slope and an intercept (you may have seen y=mx+b in algebra)

$$Y = \beta_0 + \beta_1 X$$

- Example of a trained linear regression model:
  - Target variable **Y** = House Price $
  - Explanatory variable: **X** = Num of bedrooms
  - **Regression line:** Y = 300k + 20k*X

intercept slope



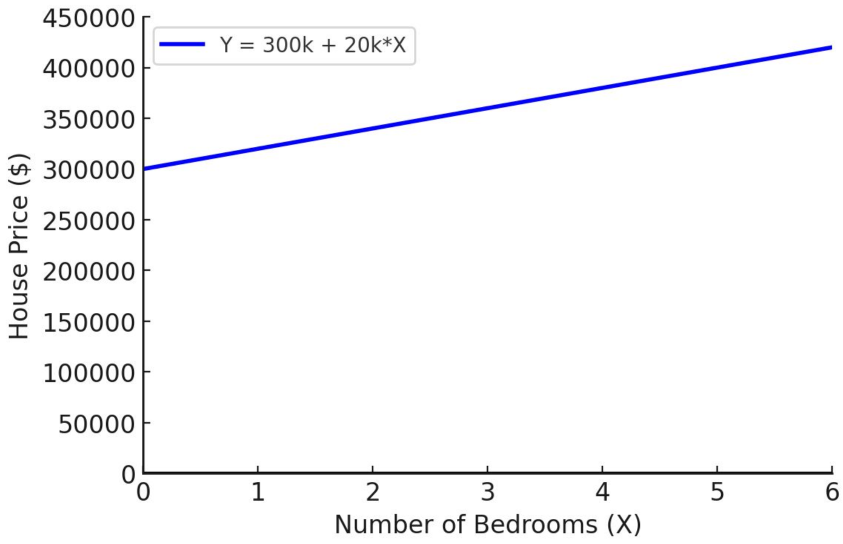
Linear Regression

- Any line is mathematically expressed as an equation with a slope and an intercept (you may have seen y=mx+b in algebra)

$$Y = \beta_0 + \beta_1 X$$

- Example of a trained linear regression model:
  - Target variable **Y** = House Price $
  - Explanatory variable: **X** = Num of bedrooms
  - **Regression line:** Y = 300k + 20k*X

intercept slope



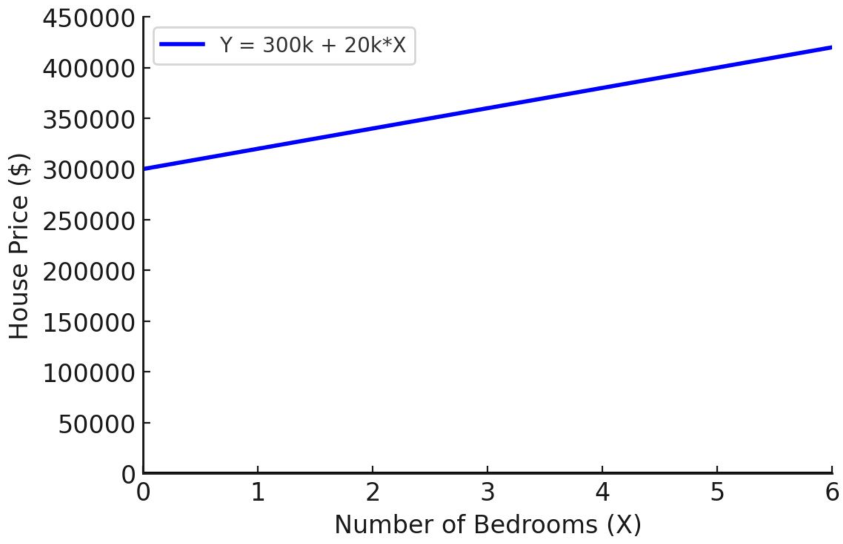
Linear Regression

- Any line is mathematically expressed as an equation with a slope and an intercept (you may have seen y=mx+b in algebra)

$$Y = \beta_0 + \beta_1 X$$

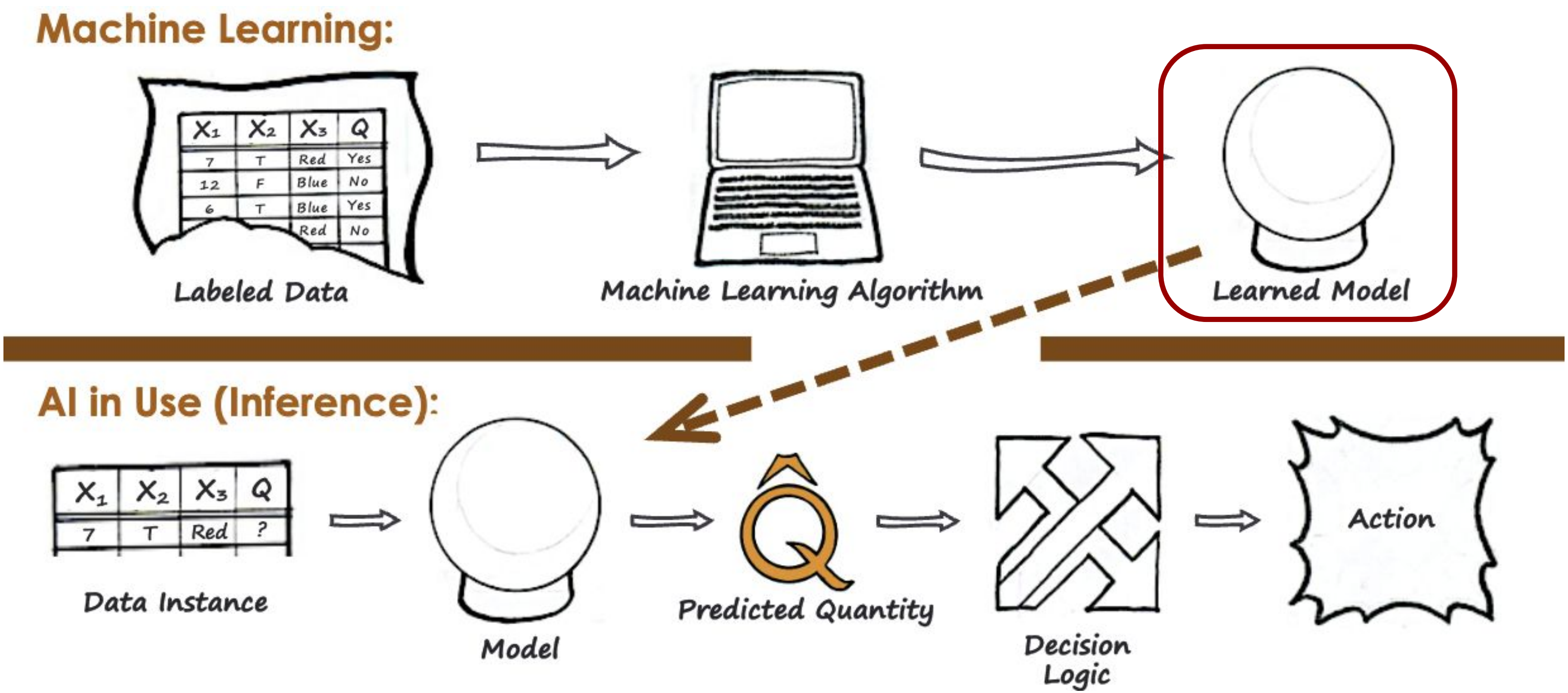
- Example of a trained linear regression model:
  - Target variable **Y** = House Price $
  - Explanatory variable: **X** = Num of bedrooms
  - **Regression line:** Y = 300k + 20k*X

intercept slope

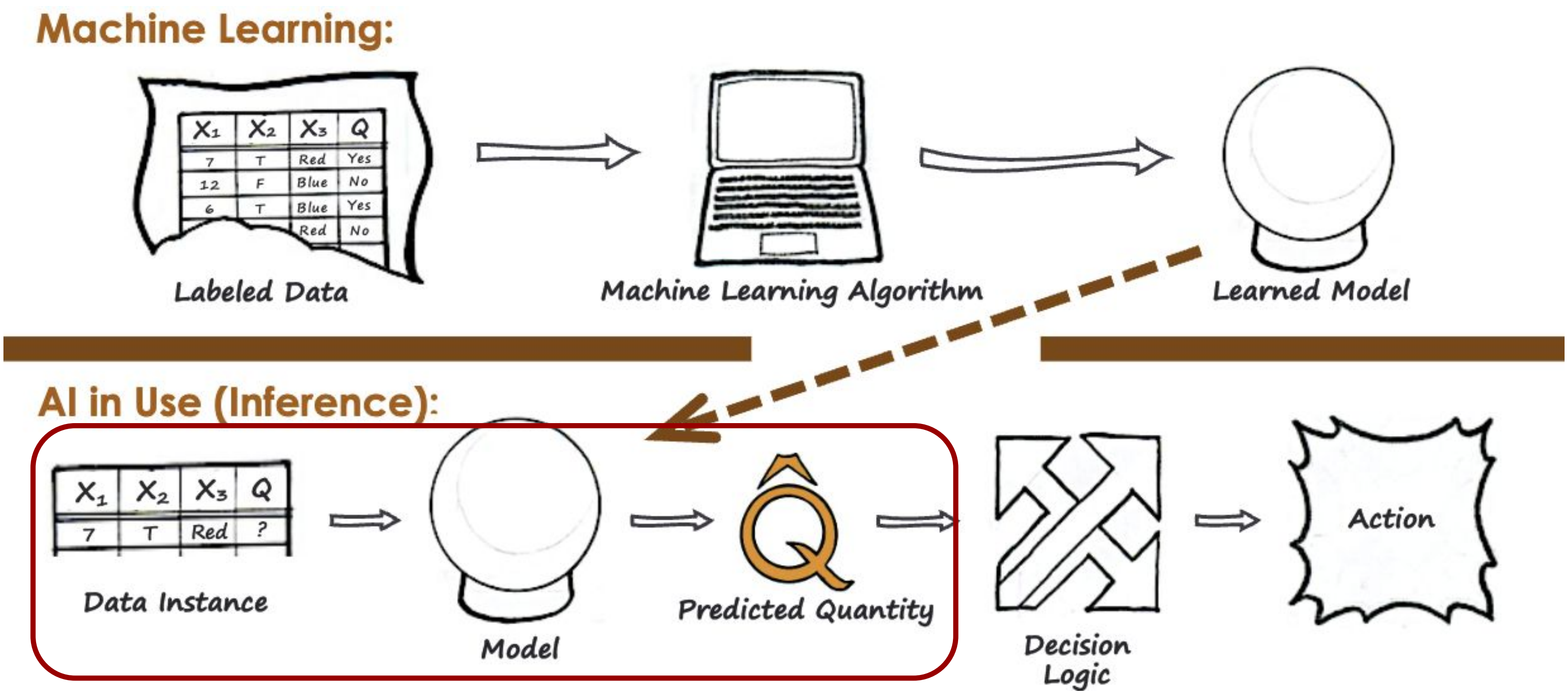


  - **Inference**: A house has 3 bedrooms, what price does the model predict?

The Predictive Analytics Flow



The Predictive Analytics Flow



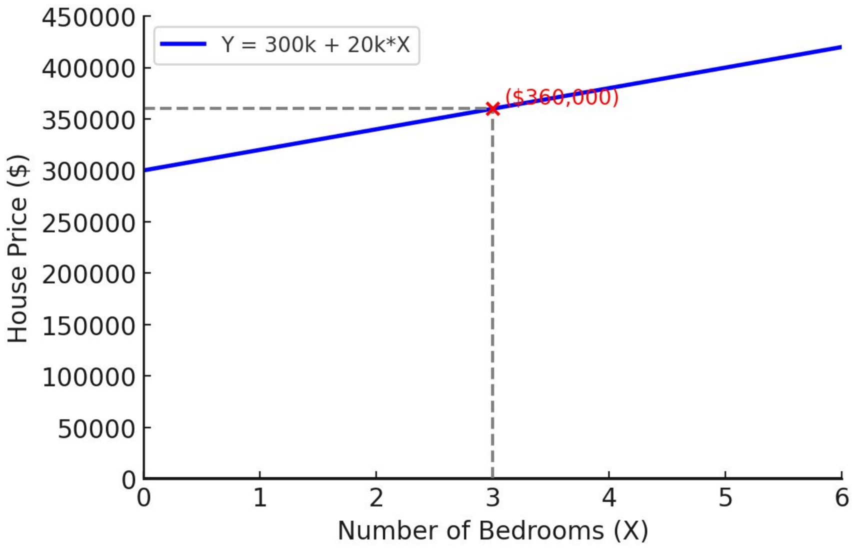
Linear Regression

- Any line is mathematically expressed as an equation with a slope and an intercept (you may have seen y=mx+b in algebra)

$$Y = \beta_0 + \beta_1 X$$

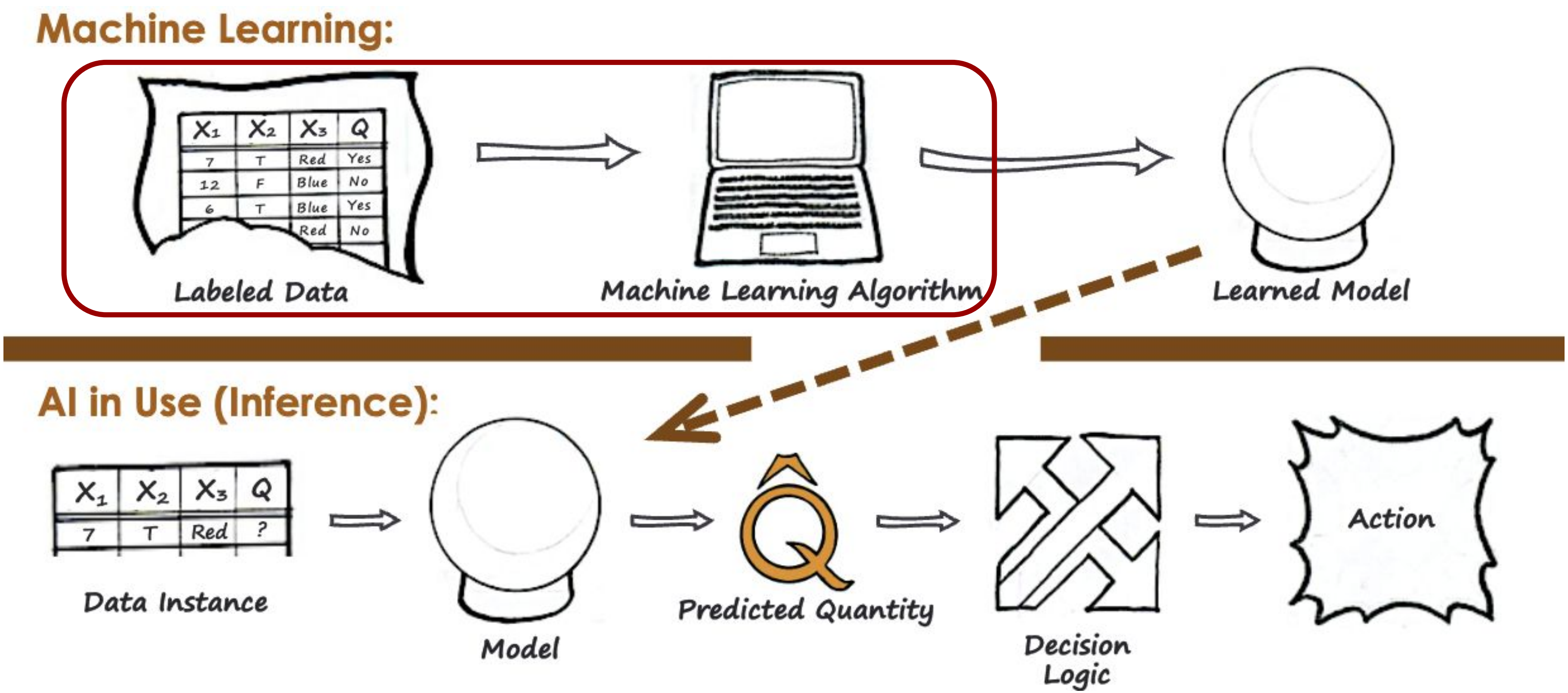
- Example of a trained linear regression model:
  - Target variable **Y** = House Price $
  - Explanatory variable: **X** = Num of bedrooms
  - **Regression line:** Y = 300k + 20k*X

intercept slope



  - **Inference**: A house has 3 bedrooms, what price does the model predict?

How do we train the model?

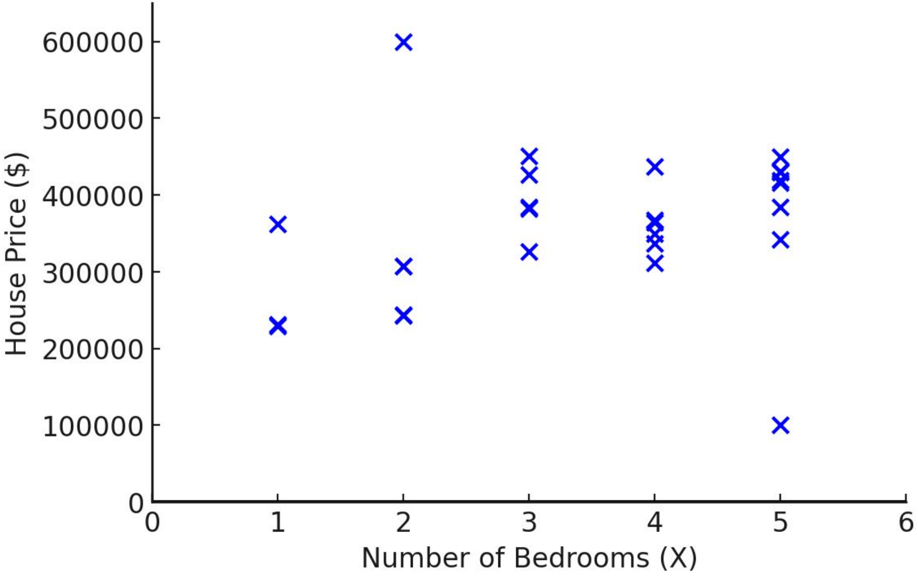


Training

- How do we **find this line that fits the data the best?**
  - Or, how do we find the parameters (the intercept and the slope?)

$$\boxed{Y = \beta_0 + \beta_1 X}$$

  - Changing the parameters, the line moves.
  - We need a measure of ‘**goodness of fit**’. How good or bad the predictions are.

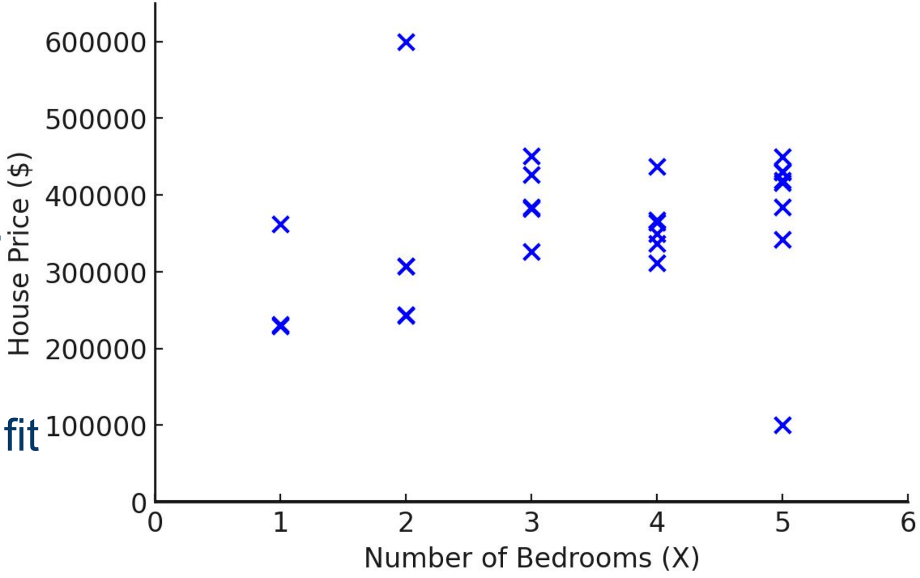


Training

- How do we **find this line that fits the data the best?**
  - Or, how do we find the parameters (the intercept and the slope?)

$$Y = \beta_0 + \beta_1 X$$

  - Changing the parameters, the line moves.
  - We need a measure of '**goodness of fit**'. How good or bad the predictions are.



- Use '**residuals**'
  - Residuals are the **errors** from the model fit
  - Residual = predicted value - actual value

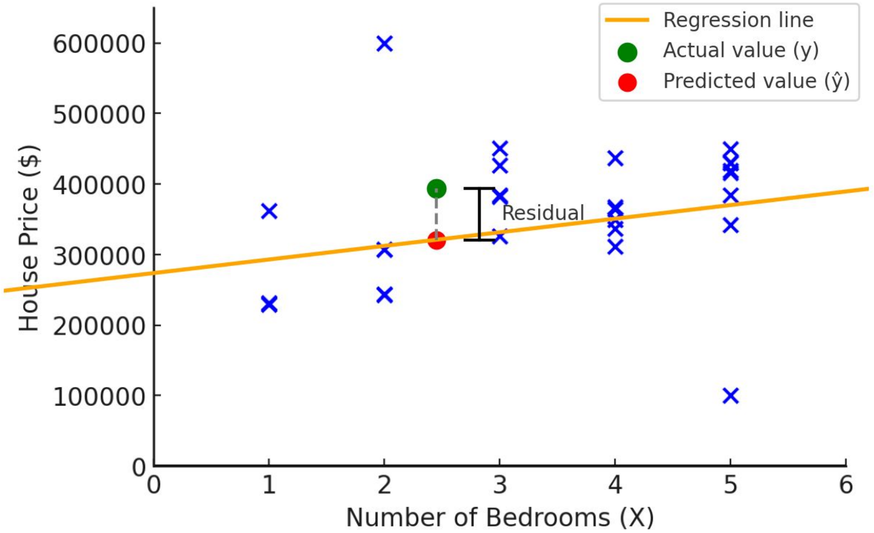
$$e_i = \hat{Y} - Y$$

Training

- How do we **find this line that fits the data the best?**
  - Or, how do we find the parameters (the intercept and the slope?)

$$Y = \beta_0 + \beta_1 X$$

  - Changing the parameters, the line moves.
  - We need a measure of '**goodness of fit**'. How good or bad the predictions are.



- Use '**residuals**'
  - Residuals are the **errors** from the model fit
  - Residual = predicted value - actual value

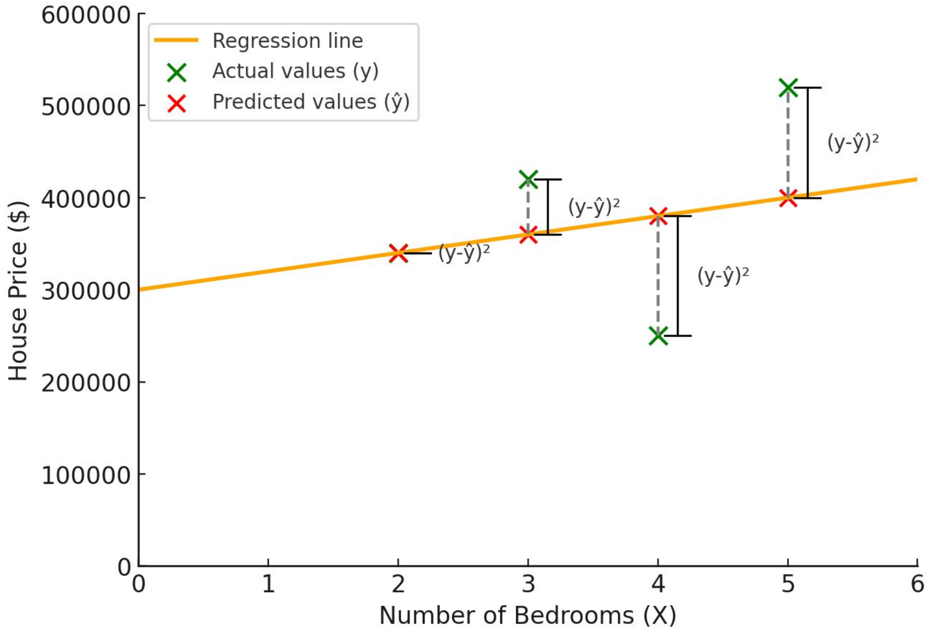
$$e_i = \hat{Y} - Y$$

Training

- The **least squares regression line** is the line that minimizes the **sum of the squared residuals.**

$$e_1^2 + e_2^2 + \ldots + + e_n^2$$

- After /n: **Mean Squared Error**
  - We call this the **loss function** of linear regression.
  - Why squared?



** Link to the interactive html*
** Link to my chatgpt history that produced the graphs and interaction html*

Training

- All ML algorithms, including Neural Networks, have **loss functions**.
- Training: **Minimizing the loss function (errors)**
  - There're optimization algorithms that could do this minimizing procedure efficiently.

Multiple Linear Regression

- What if I have more features?

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X2 + \beta_3 X3 + \ldots.$$

Trained model: House Price = 300K + 20k* Num bedrooms
+ 1k* Num bathrooms
+ 100k* If renovated
+ 100 * SQFT
+ …

Add some non-linearity

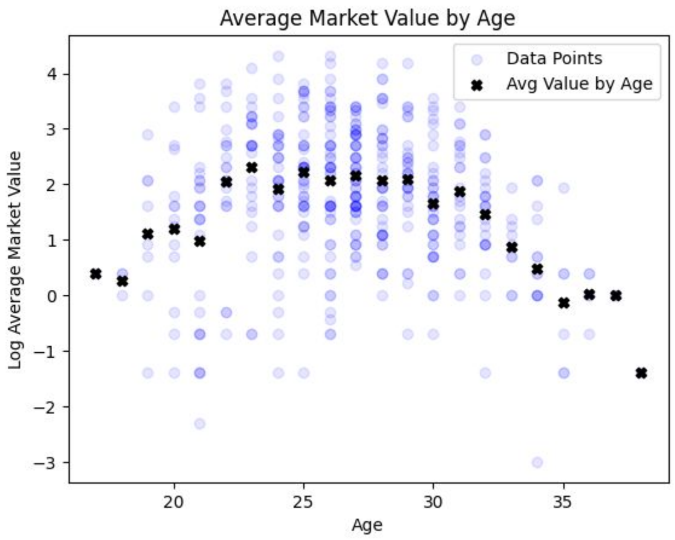
- But still linear regressions!

$$Y = \beta_0 + \beta 1_X 1 + \beta_2 log(X2) + \beta_3 X_3^2 + \dots$$

Trained model: House Price = 300K + 20k* Num bedrooms
+ 1k* log(Num bathrooms)
+ 100k* If renovated
+ 100 * SQFT
- 3*SQFT^2

Add some non-linearity

- English premier soccer players; Value ~ age



- But if you calculate a variable that's quadratic age, so the x-axis is for quadratic age, the line is still straight.

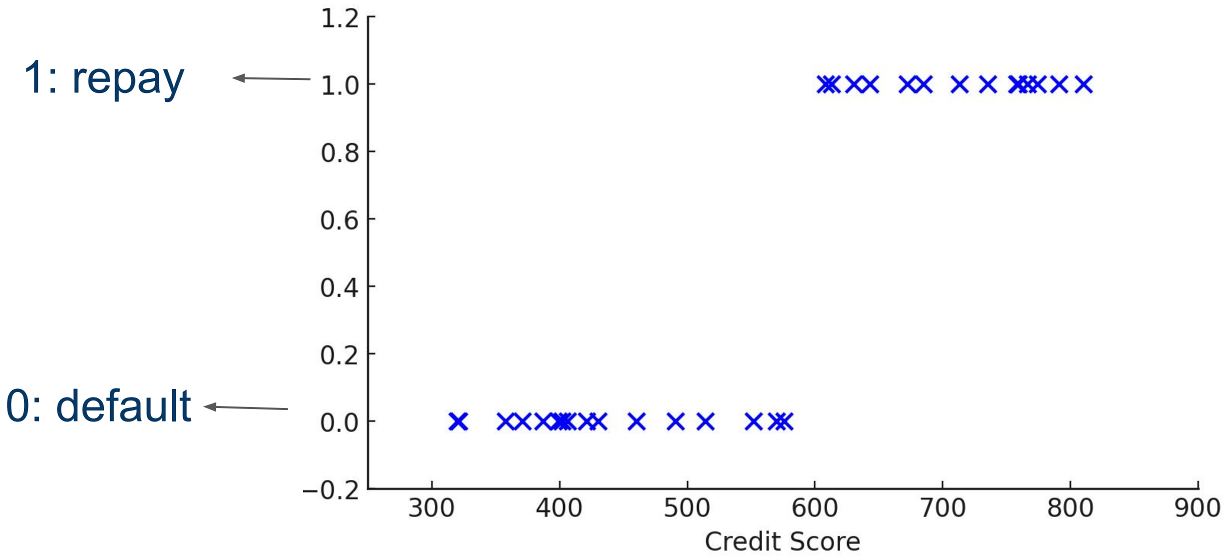
Logistic Regression

- Actually a technique for a **Classification problem**.
  - With a binary/ categorical target variable. (e.g. a customer, churn or not?)

  - If a customer will purchase a product?
  - If a stock is going up or down tomorrow?
  - If I’d like to hire someone, is he/she going to accept?
  - If my company want to enter a new market, will it succeed?
  - There’re a bunch of potential clients, who do I reach out first?

Logistic Regression

- Actually a technique for a **Classification problem**.
  - With a binary/ categorical target variable. (e.g. a customer, churn or not?)
  - E.g. Credit approval. Need to decide if an applicant will repay/default based on credit score.



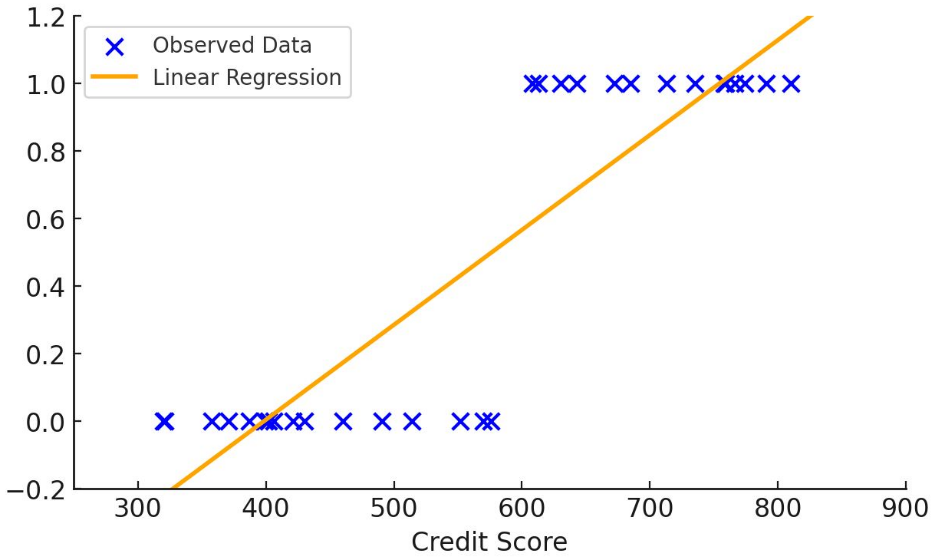
Logistic Regression

- Actually a technique for a **Classification problem**.
  - Typically with a binary target variable. (e.g. a customer, churn or not?)
  - E.g. Credit approval. Need to decide if an applicant will repay/default based on credit score.

- Let's try the linear regression

$$\hat{y} = -1.12 + +0.002 \cdot \mathrm{CreditScore}$$

  - It outputs real numbers, not binary
  - It can predict something <0 or >1
  - Doesn't make sense for classification



Logistic Regression

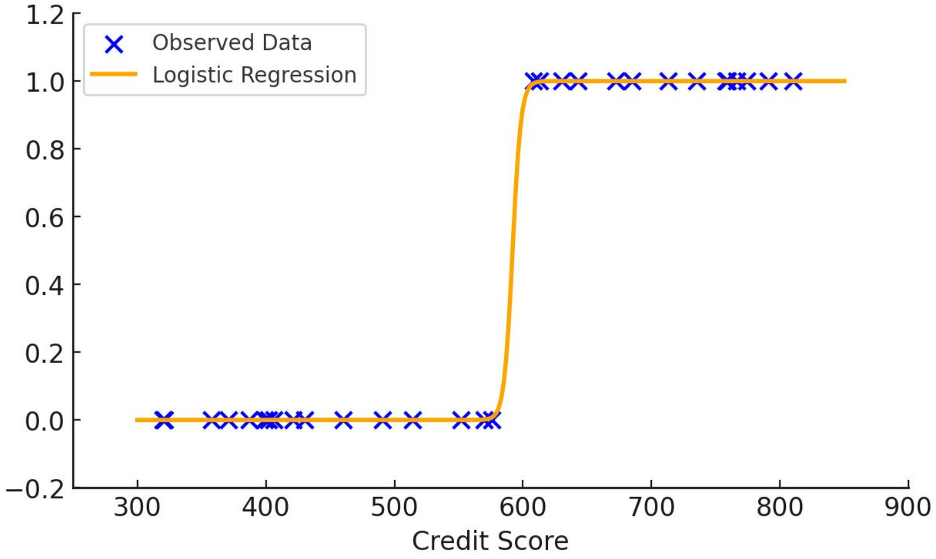
- Actually a technique for a **Classification problem.**
  - Typically with a binary target variable. (e.g. a customer, churn or not?)
  - E.g. Credit approval. Need to decide if an applicant will repay/default based on credit score.

- So, **logistic regression model**:

$$\eta(x) = \beta_0 + \beta_1 x$$

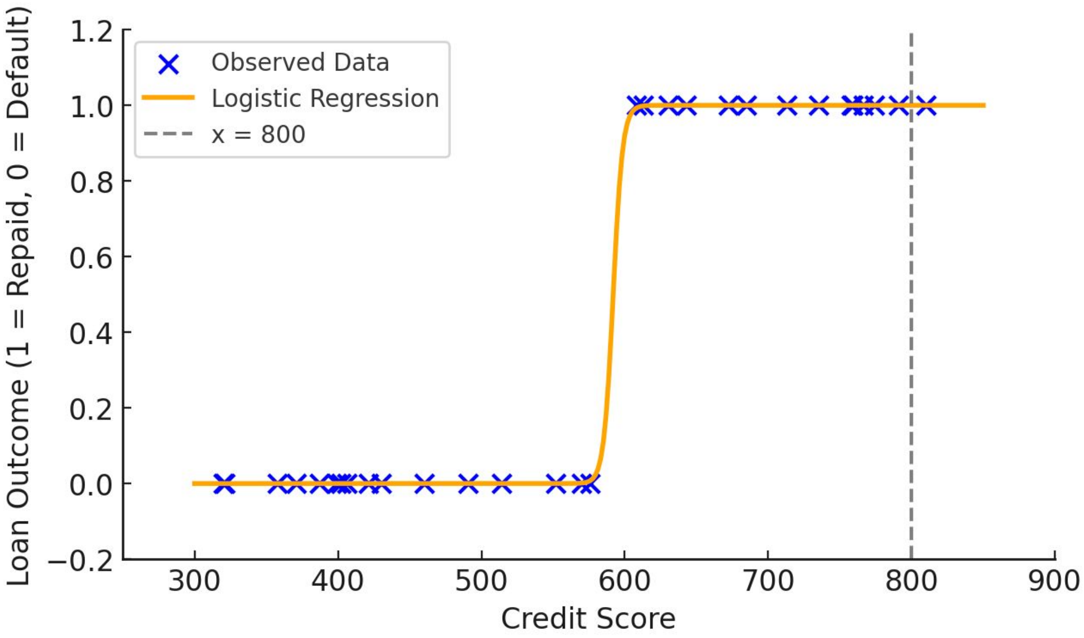
$$Pr(Y=1|X) = Sigmoid(\eta(x)) = \frac{1}{1+e^{-\eta(x)}}$$

  - The sigmoid function: map any number to a value between 0 and 1
  - So we could **interpret the output as probabilities**!



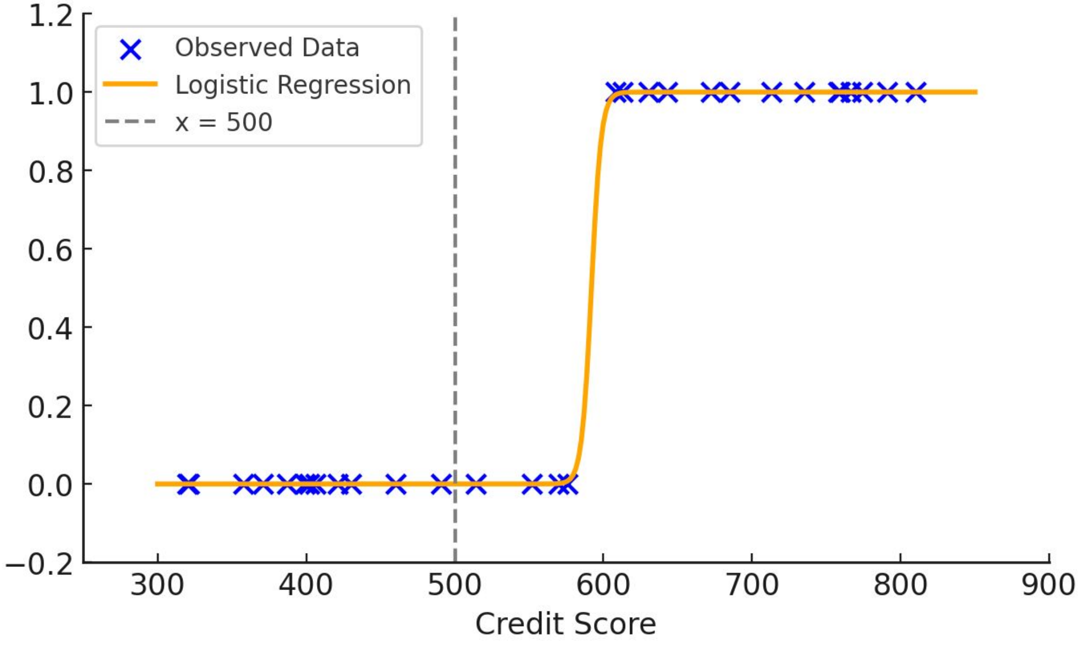
Logistic Regression

For someone with a credit score of 800, chance of repaid loan?



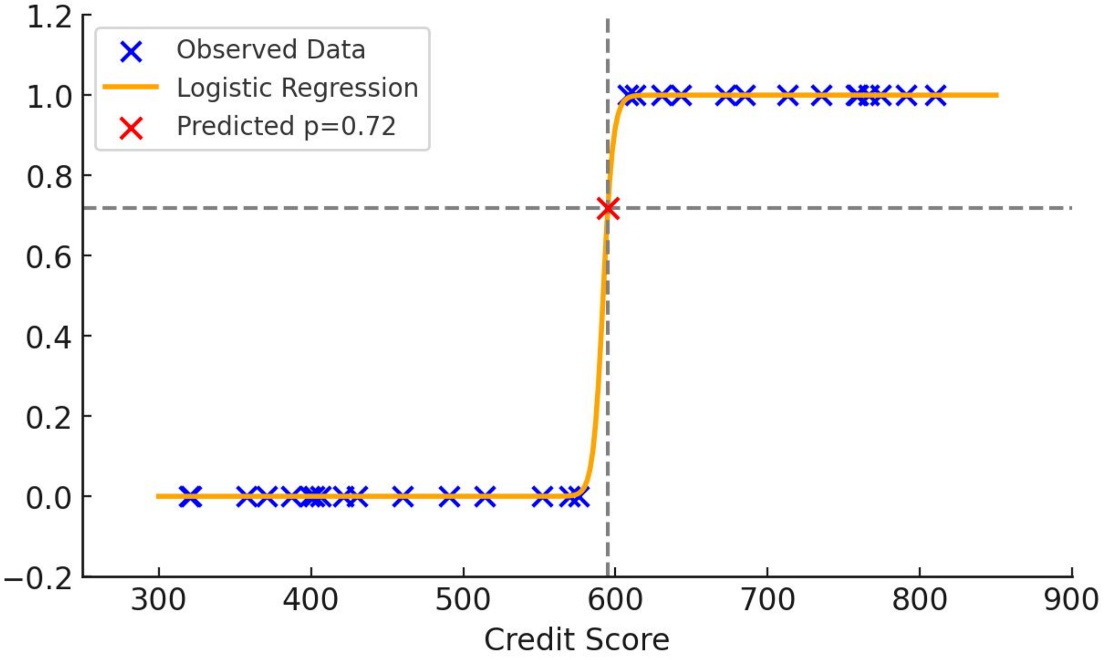
Logistic Regression

For someone with a credit score of 500, chance of repaid loan?



Logistic Regression

For someone with a credit score of 595, chance of repaid loan?



Logistic Regression

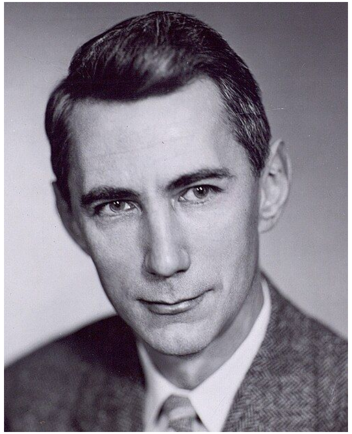
- So, how do we find this logistic regression line?
  - What're the errors -> loss function?

Cross entropy!

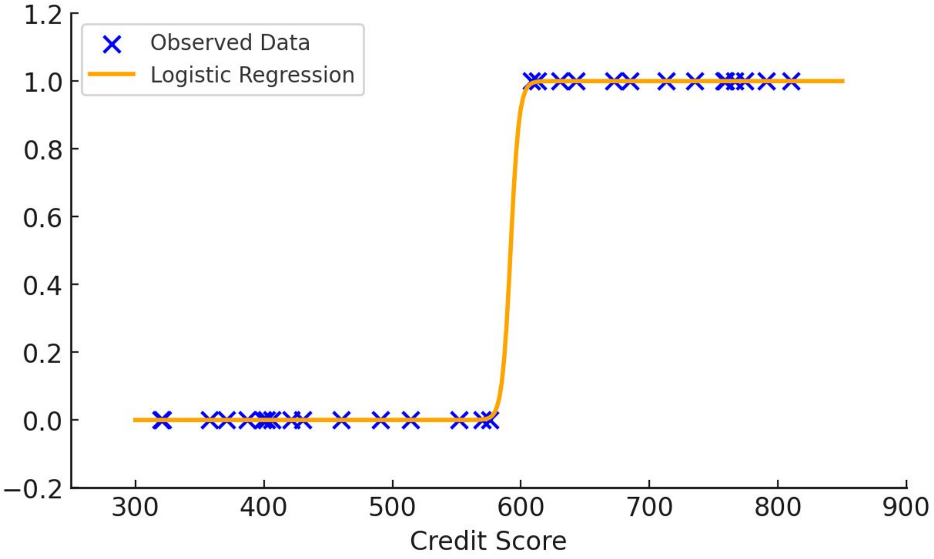
$$\mathcal{L}(\beta_0, \boldsymbol{\beta}) = -\sum_{i=1}^{n} \left[y_i \log p_i \; + \; (1-y_i) \, \log(1-p_i) \right]$$

$$\eta(x) = \beta_0 + \beta_1 x$$

$$Pr(Y=1|X) = Sigmoid(\eta(x)) = \frac{1}{1+e^{-\eta(x)}}$$

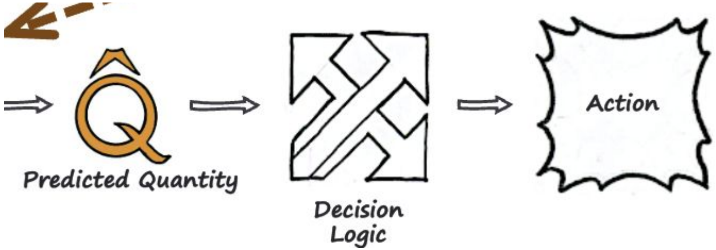


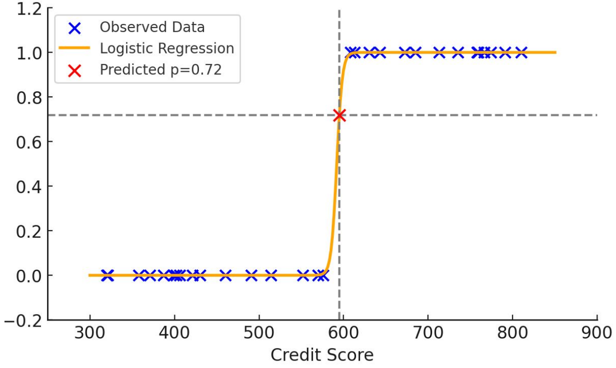
- Cross-entropy heavily punishes confident wrong predictions.



We still need to classify..

- Need to choose a cutoff/ threshold:

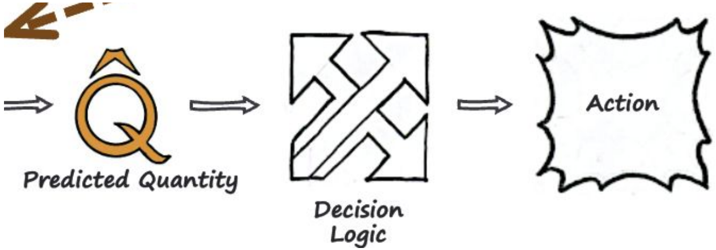


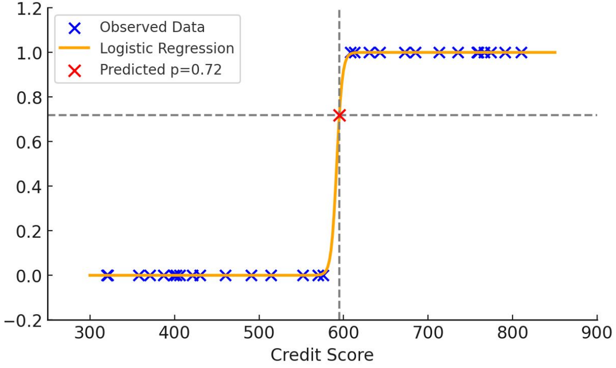


- If predicted Pr(repay) > **threshold**, approve loan

We still need to classify..

- Need to choose a cutoff/ threshold:



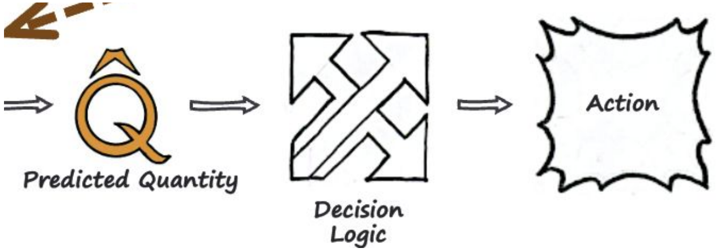


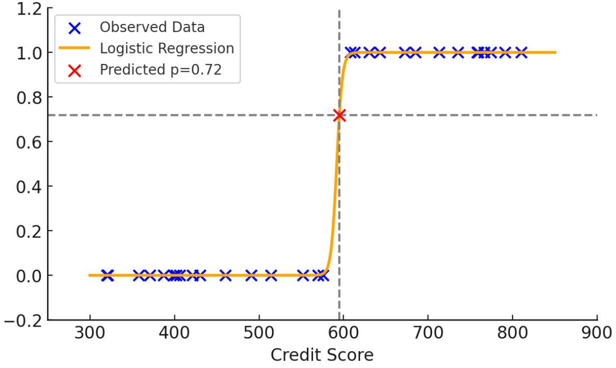
- If predicted Pr(repay) > **threshold**, approve loan
  = 0.6

Customer	Credit Score	Predicted Pr(Repay)	Decision (Threshold=0.6)
A	520	0.42	
B	580	0.59	
C	640	0.72	
D	710	0.81	
E	820	0.93	

We still need to classify..

- Need to choose a cutoff/ threshold:



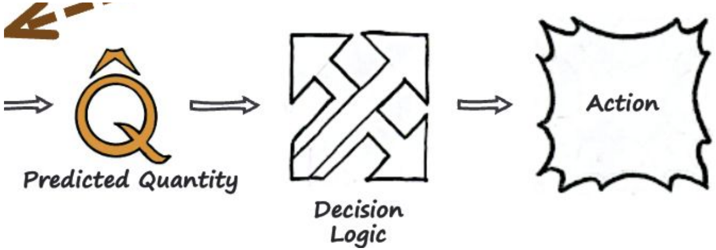


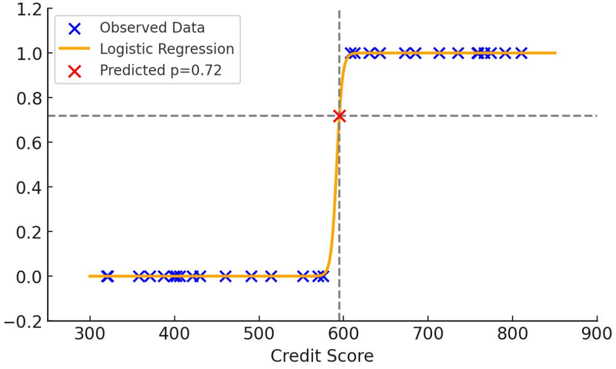
- If predicted Pr(repay) > **threshold**, approve loan **= 0.8**

Customer	Credit Score	Predicted Pr(Repay)	Decision (Threshold=0.6)
A	520	0.42	No
B	580	0.59	No
C	640	0.72	Approve
D	710	0.81	Approve
E	820	0.93	Approve

We still need to classify..

- Need to choose a cutoff/ threshold:





- If predicted Pr(repay) > **threshold**, approve loan
  = 0.5

Customer	Credit Score	Predicted Pr(Repay)	Decision (Threshold=0.6)
A	520	0.42	No
B	580	0.59	No
C	640	0.72	Approve
D	710	0.81	Approve
E	820	0.93	Approve

A

- Moving down the threshold -> Giving out more loans, but more likely to default.

We still need to classify..

- Need to choose a cutoff/ threshold (Well this should've been the first thing to do before model building. This is the busines understanding part!):

Customer	Credit Score	Predicted Pr(Repay)	Decision (Threshold=0.6)	Ground Truth (Repay=1, Defaul
A	520	0.42	No	0
B	580	0.59	No	1
C	640	0.72	Approve	0
D	710	0.81	Approve	1
E	820	0.93	Approve	1

- Business understanding:
  If approve, but defaulted, loss = Loan value;
  Repaid, profit = interest.
  If not approve, no gain no loss
- Decision logic: E[profit|approve] = Pr(repaid)*interest - Pr(default)*loan value > 0

Classification task

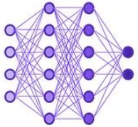
Is LLM modeling a classification task?

Classification task

Is LLM modeling a classification task?

The cat likes to sleep in the

Input



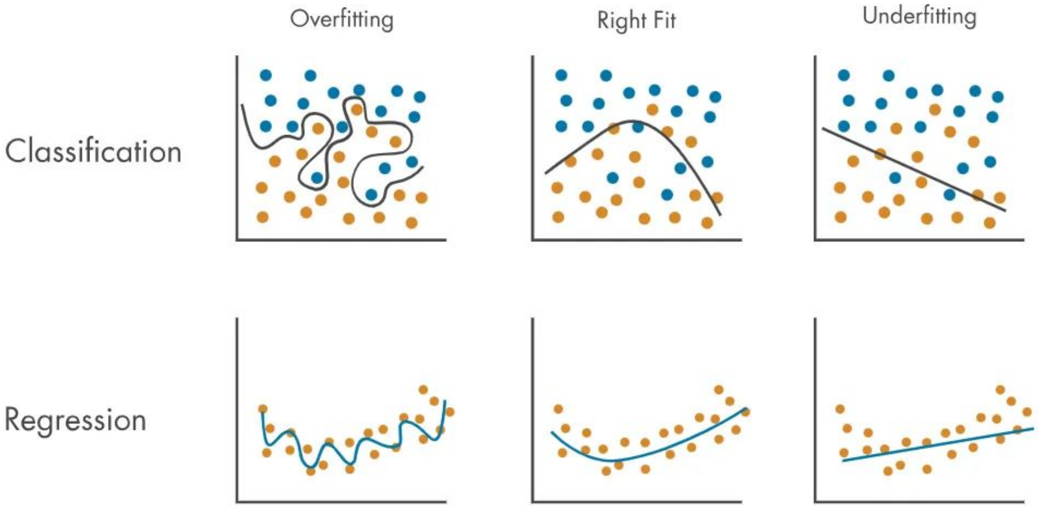
Neural Network (LLM)

Word	Probability
ability	0.002
bag	0.071
box	**0.085**
...	...
zebra	0.001

Output

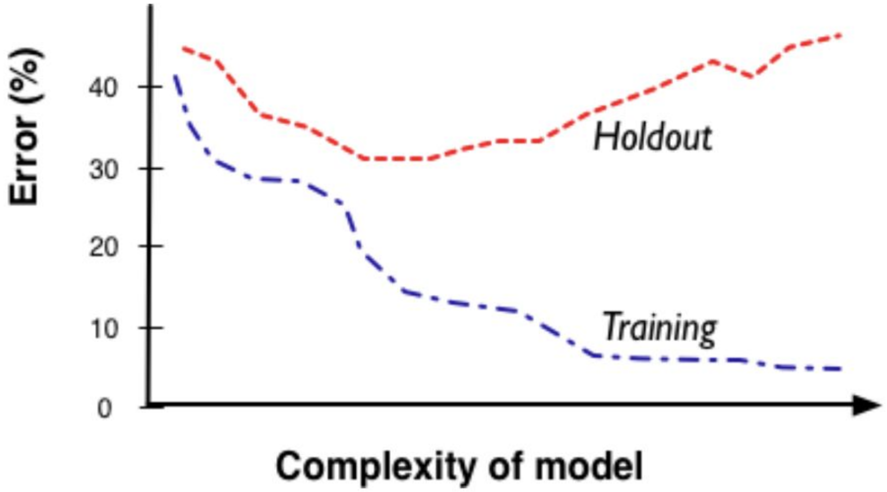
Evaluation

- Why do we need to evaluate model performances?
  - Necessary for ensuring that the models can make accurate predictions on **new, unseen** data!
- **Overfitting** happens when a model learns the training data too well (learning the random noises and quirks)
  - it performs great on the training set but **poorly on new, unseen data**.



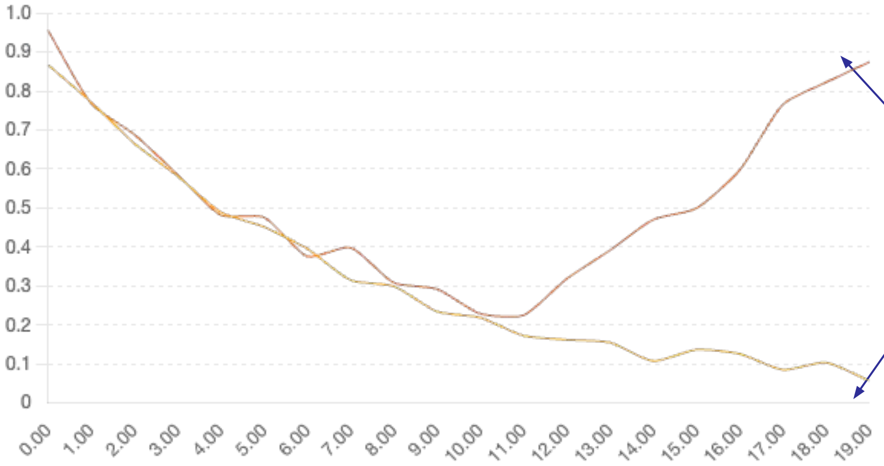
Evaluation

- More **complexity** allows us more freedom and flexibility to fit the messy realities
  - Models always get better (as measured on training set) with more data or more complexity. BUT higher complexity runs the risk of **overfitting** data.
- So we need holdout data to optimize generalizability.
  - Basically, we break our dataset into a **training set**, and **a validation set**.
  - And evaluate the performance on the validation set.



Evaluation

- More **complexity** allows us more freedom and flexibility to fit the messy realities
  - Models always get better (as measured on training set) with more data or more complexity. BUT higher complexity runs the risk of **overfitting** data.
- So we need holdout data to optimize generalizability.
  - Basically, we break our dataset into a **training set**, and **a validation set**.
  - And evaluate the performance on the validation set.



Overfitting: model is too complex: the model fits great on the training data but doesn't generalize to holdout data because it is "memorizing" the training data.

Complexity in Linear/ Logistic Regressions

- For regressions models, complexity comes in multiple forms – often called the "dimensionality" of the model.
  - **More attributes** means more complex relationships
  - **Categorical variables** can explode dimensionality.

Adding attributes might make our model better!

But now we've added complexity and might be overfitting

There are real-world scenarios where we may want to explore hundreds, thousands, even MILLIONS of attributes

- Financial models with time based attributes

Overfitting

Does overfitting happen with LLMs too?

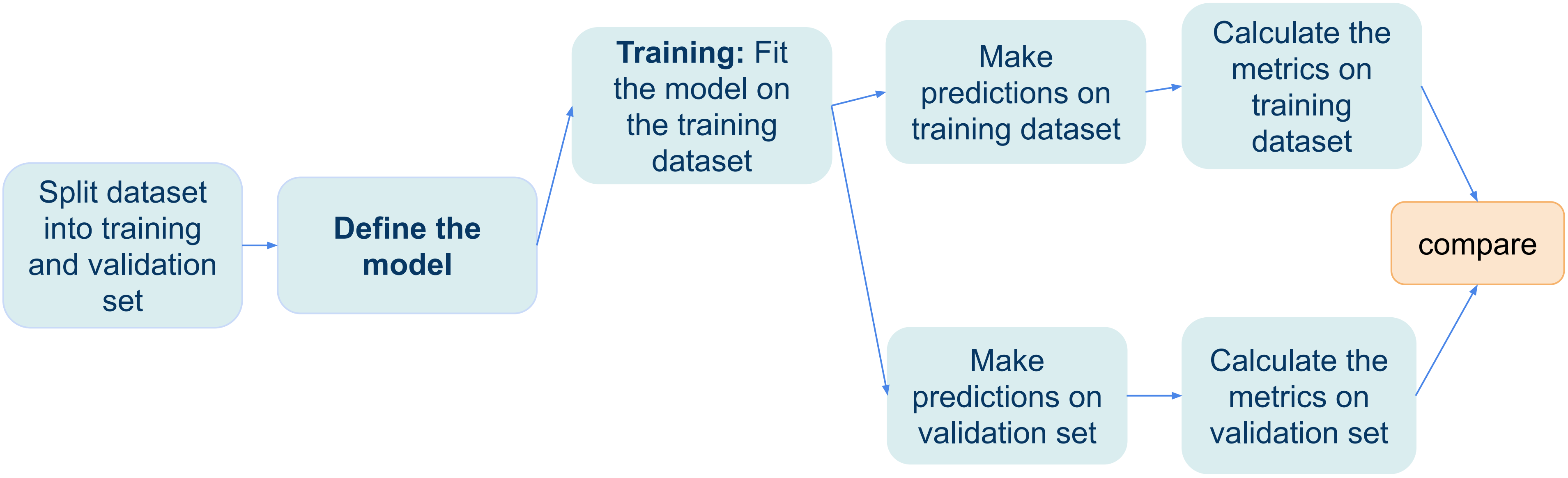
Yeah! But, wait a few weeks.

Evaluation Metrics

- For regression models:
  - What did we want to minimize? The errors! Mean Squared Error (MSE)!

- For classification tasks:
  - What did we want to minimize?
    - Cross entropy
    - Prediction accuracy
    - Payoffs (Cost of wrongly classified points have costs? [More on this on Thursday])

The evaluation workflow



Linear Regression - Model building, training, evaluation

- **Define the model:** What does the model look like?

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X2 + \beta_3 X3 + \ldots.$$

- **Training:** How do we find the model that's best fit to data (on the training set)?
  - Find the parameters that minimizes the mean squared error.

$$e_1^2 + e_2^2 + \ldots + + e_n^2$$

$$e_i = \hat{Y} - Y$$

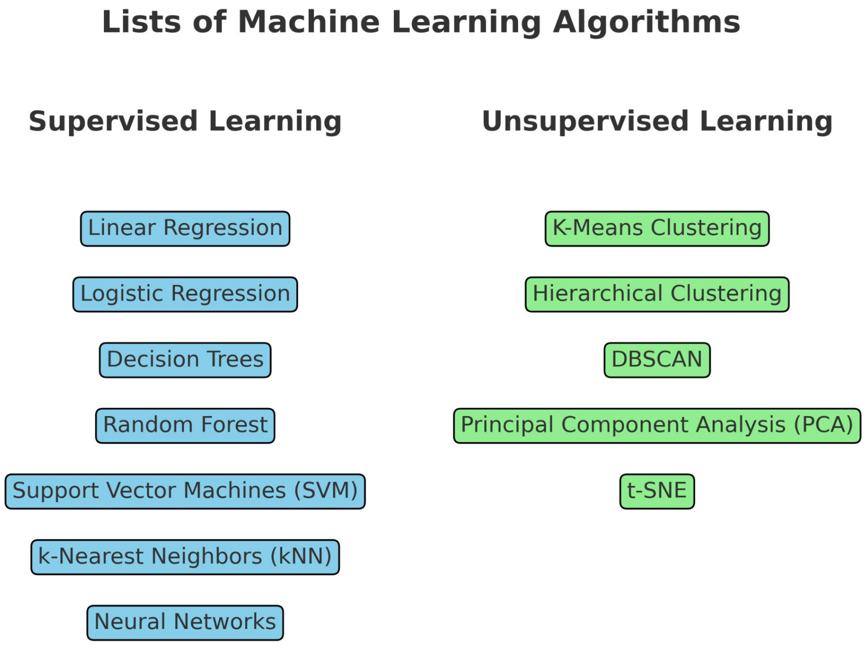
- **Evaluation:** How do we know if the model performs well on new data?
  - We have the trained model parameters, just plug in feature values to make predictions on the validation set, and measure the MSE.
  - Check if the model fit is significantly worse than on the training set.

Classification - Model building, training, evaluation

- **Define the model:**
  Sigmoid(linear) – What're the features?

- **Training:**
  - Loss function to minimize: Cross entropy
  - So the model optimize for that and find the parameters.

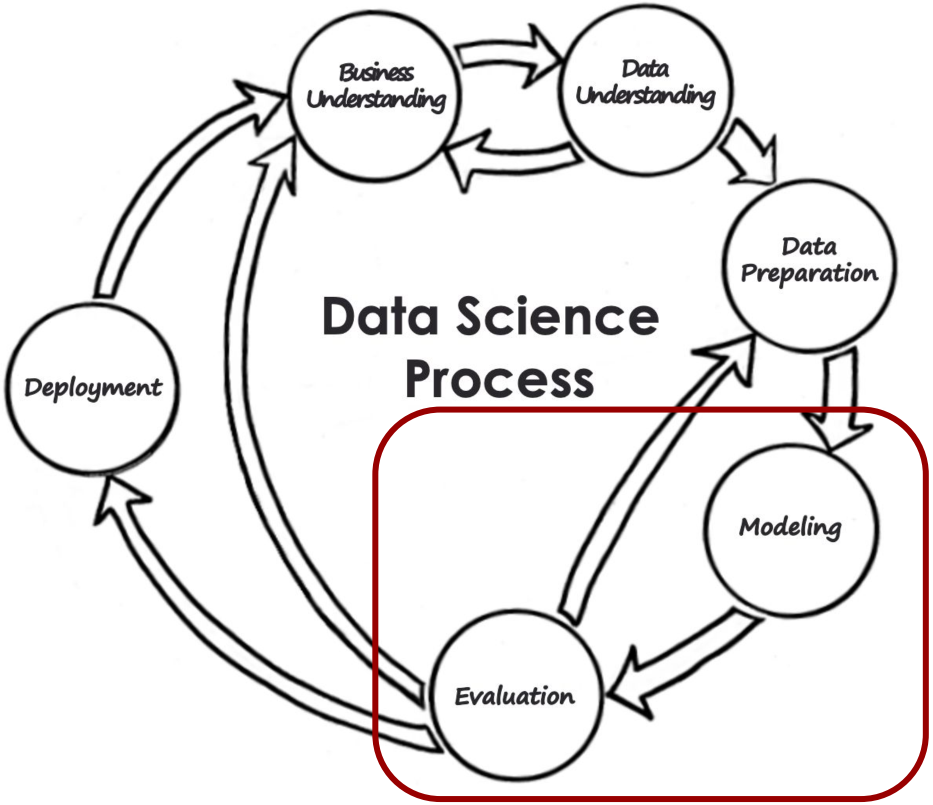
- **Evaluation:**
  - We have the trained model parameters, just plug in feature values to make predictions on the validation set, and measure the metric.
  - Check he prediction accuracy.
  - When it involves decision making, calculate the overall costs/benefits.



The ML course will walk you through these algorithms.

** Asked chatgpt for a list of supervised/unsupervised ML algorithms and give me a .png*

The Data Science Process



Great! We're well equipped to learn about Neural Networks!

More Evaluation Metrics for Binary Classification Tasks aside from cross entropy

Confusion Matrix

- For a chosen threshold, you could calculate a confusion matrix.

predictions

true

	Churned	Nope
Churned	True positive %	False negative %
Nope	False positive%	True negative%

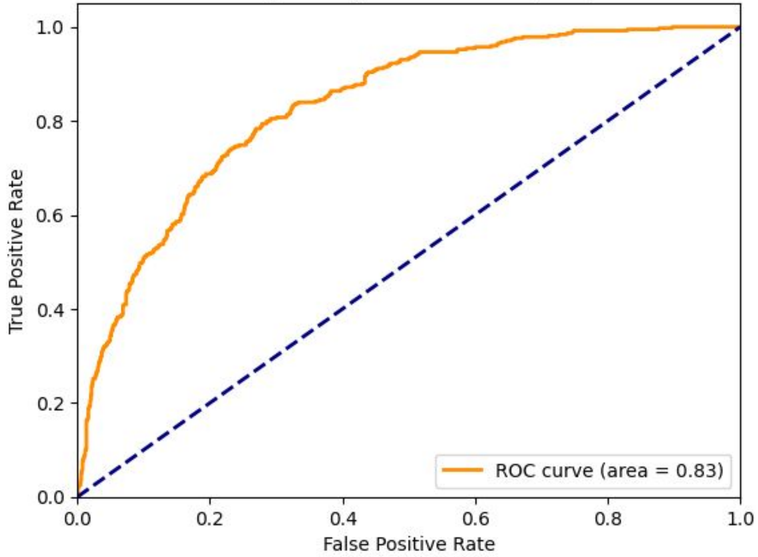
- People who actually churn, how good the model is to detect them.

- People who didn’t churn, how wrong the model is?
E.g., You’re healthy, the model says you got cancer.

ROC Curve

- For all threshold, we could calculate a confusion matrix.
  So let's put the true positive % on the y-axis, and false positive % on the x-axis
  - ROC curve: If I move the threshold up and down, how well does the model keep positives ranked above negatives?

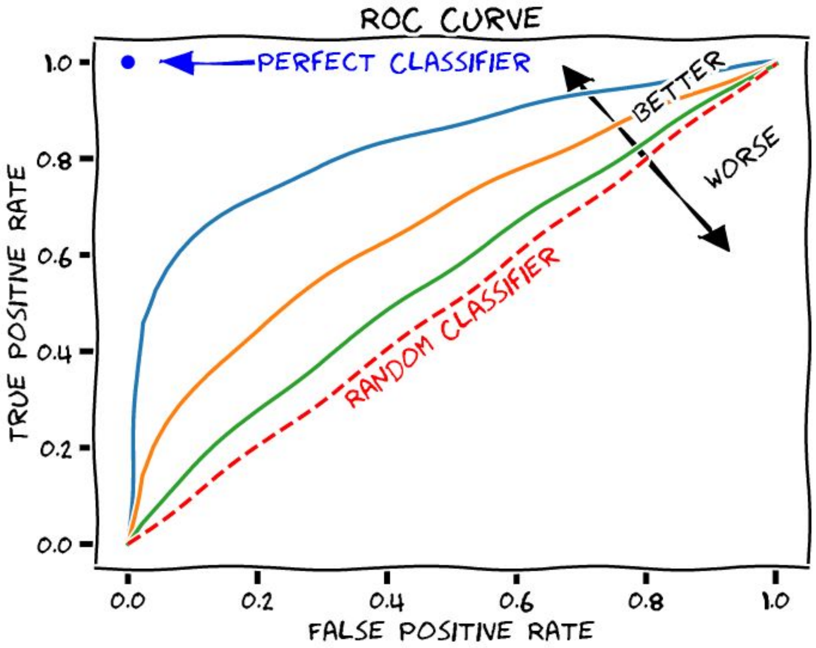
- Catching real positive



- Raising false alarms

ROC Curve

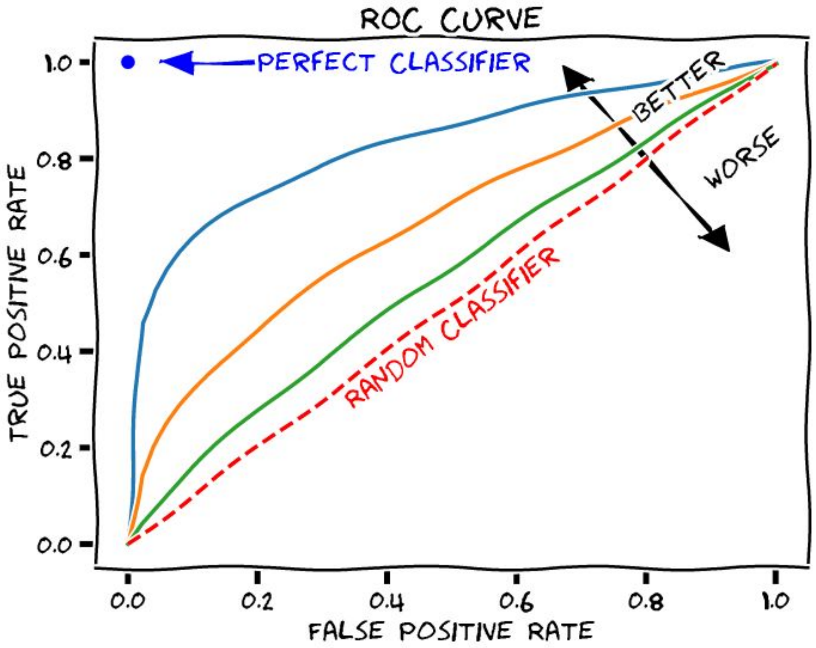
- Plot a line for each model. Closer to the y-axis, the better.
  - The model is good at rank true positives at the top (larger probabilities)
  - The model is good at separating people who churned v.s. not.



- Same amount of mistakes, more likely to get things right.

AUC

- Area under the ROC curve.
  - Larger, the better.



- Same amount of mistakes, more likely to get things right.