University of Salford MANCHESTER

Artificial Intelligent – Week 6

Dr Salem Ameen

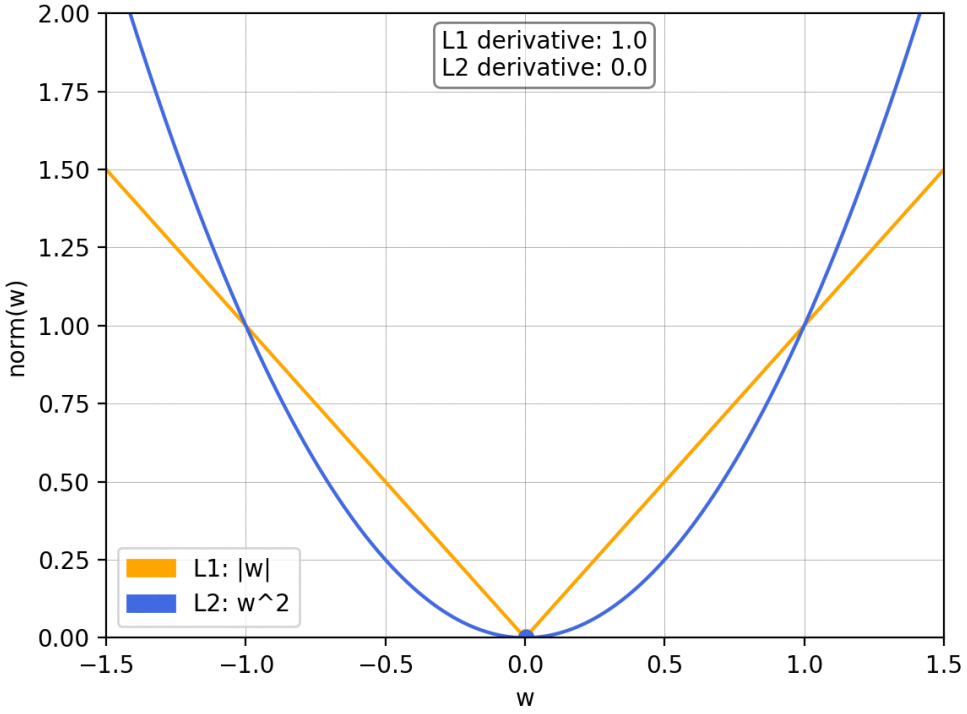
S.A.AMEEN1@SALFORD.AC.UK

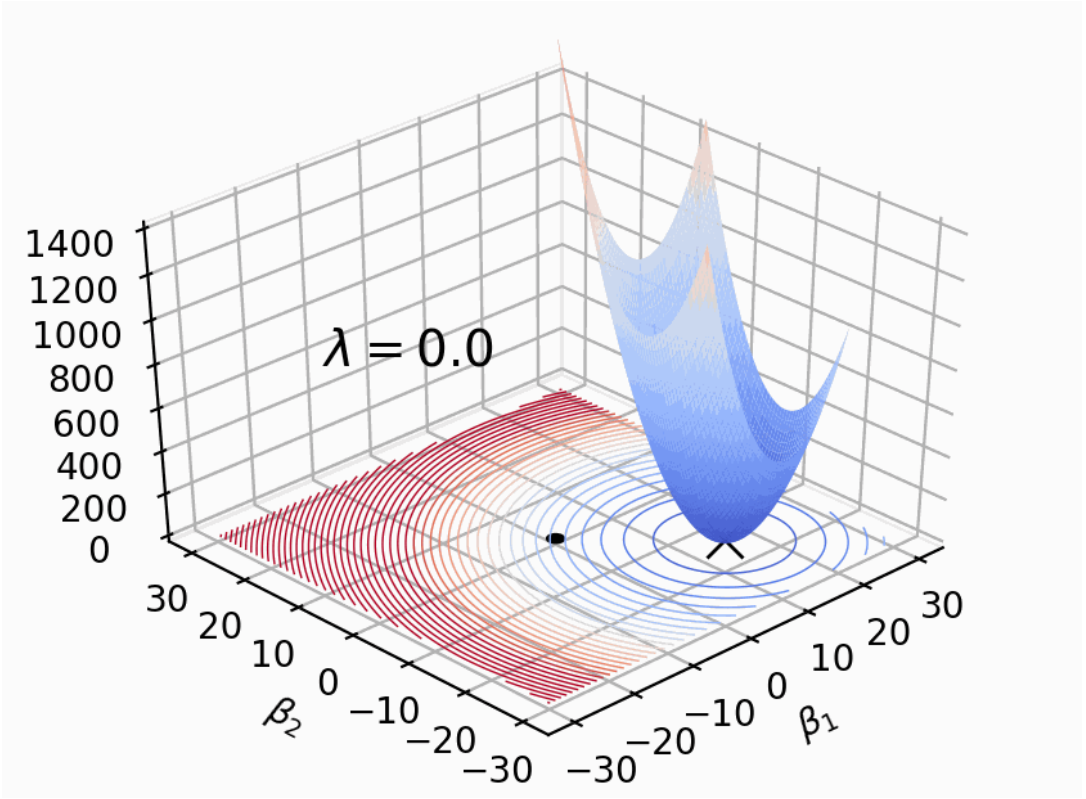
Week 6 – Regression
Last Lecture

University of Salford MANCHESTER

Loss and Regularizer	Comments
Ordinary Least Squares $\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top x_i - y_i)^2$	◦ Squared Loss ◦ No Regularization ◦ Closed form solution: ◦ $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{y}^\top$ ◦ $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ ◦ $\mathbf{y} = [y_1, \ldots, y_n]$
Ridge Regression $\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top x_i - y_i)^2 + \lambda\|w\|_2^2$	◦ Squared Loss ◦ l_2-Regularization ◦ $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbb{I})^{-1}\mathbf{X}\mathbf{y}^\top$
Lasso $\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda\|\mathbf{w}\|_1$	◦ + sparsity inducing (good for feature selection) ◦ + Convex ◦ - Not strictly convex (no unique solution) ◦ - Not differentiable (at 0) ◦ Solve with (sub)-gradient descent or SVEN
Elastic Net $\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \alpha\|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2$ $\alpha \in [0, 1)$	◦ ADVANTAGE: Strictly convex (i.e. unique solution) ◦ + sparsity inducing (good for feature selection) ◦ + Dual of squared-loss SVM, see SVEN ◦ DISADVANTAGE: - Non-differentiable

Week 6 – Regression
Last Lecture

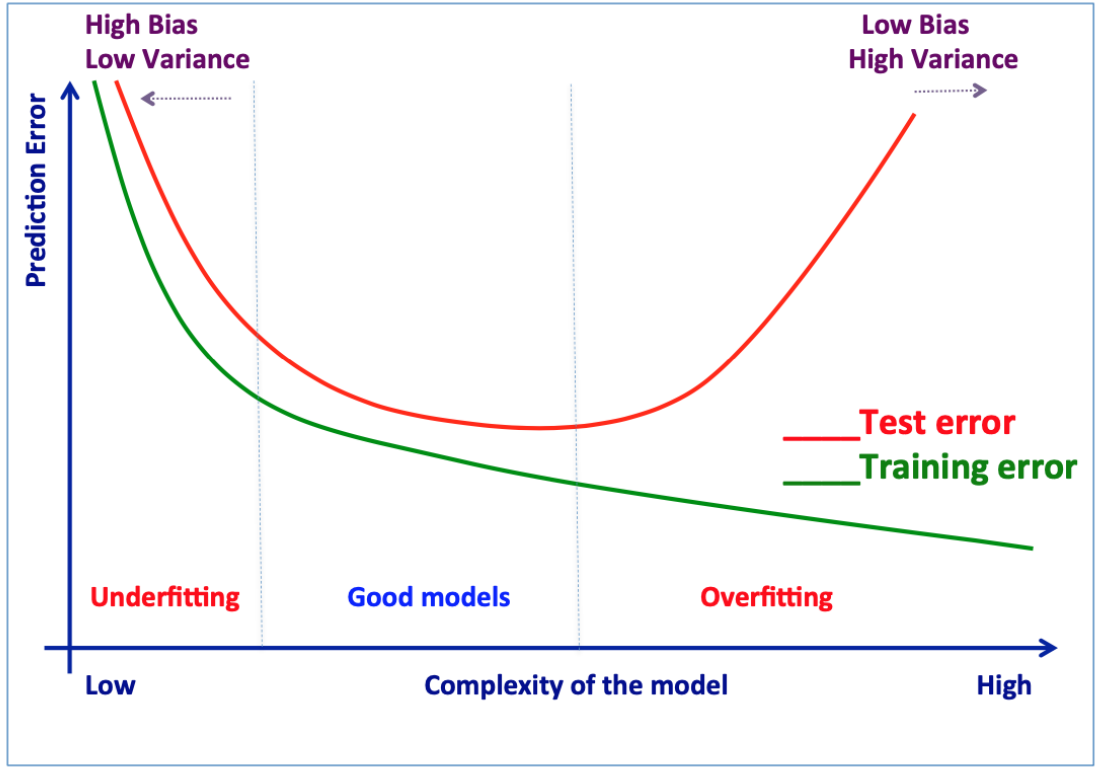




Week 6 – ML Generalization

University of Salford MANCHESTER

$$\underbrace{E_{\mathbf{x},y,D}\left[\left(h_D(\mathbf{x})-y\right)^2\right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x})-\bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y}\left[\left(\bar{y}(\mathbf{x})-y\right)^2\right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x})-\bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

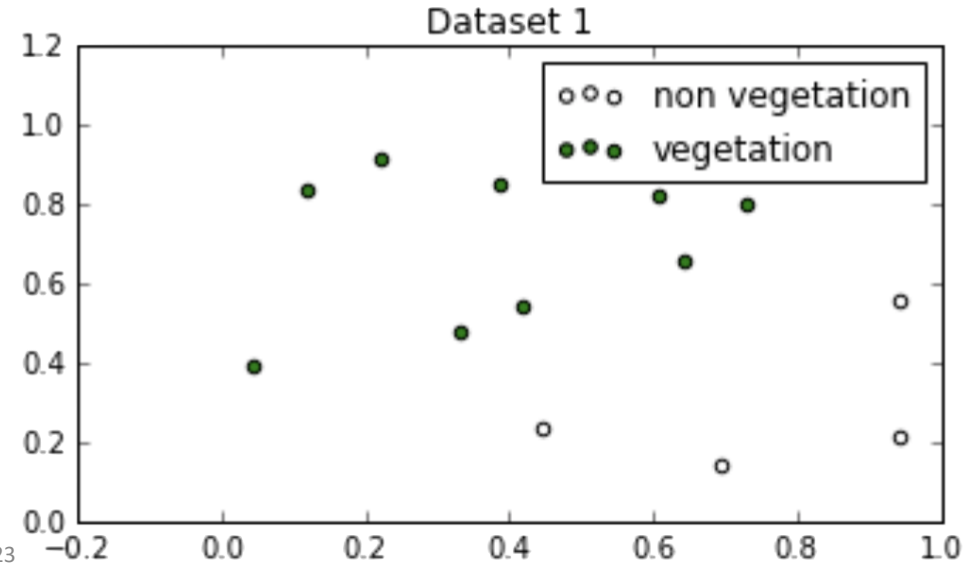


Week 6
Geometry of Data

University of Salford MANCHESTER

Recall:

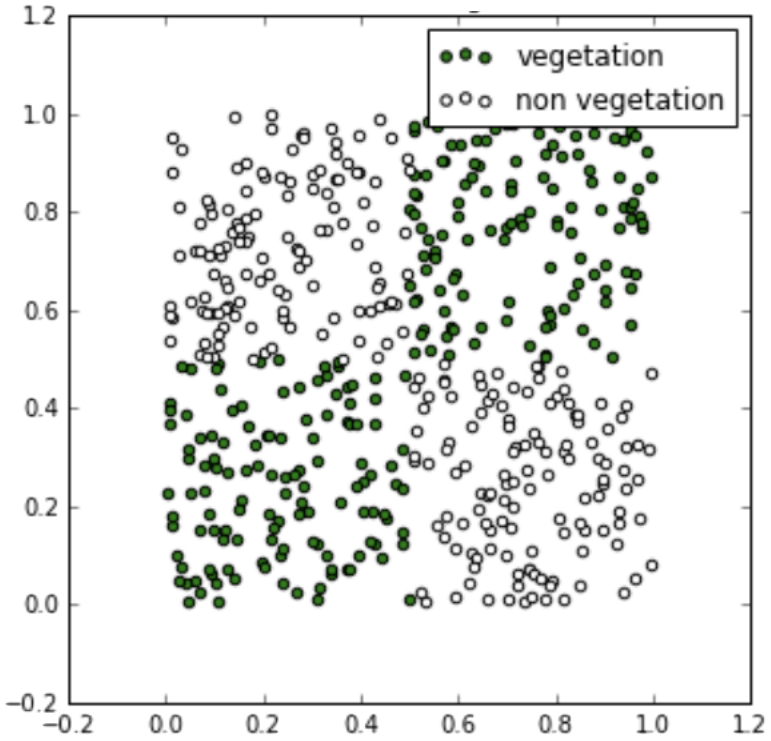
- **logistic regression** for building classification boundaries works best when:
  - the classes are well-separated in the feature space
  - have a nice geometry to the classification boundary

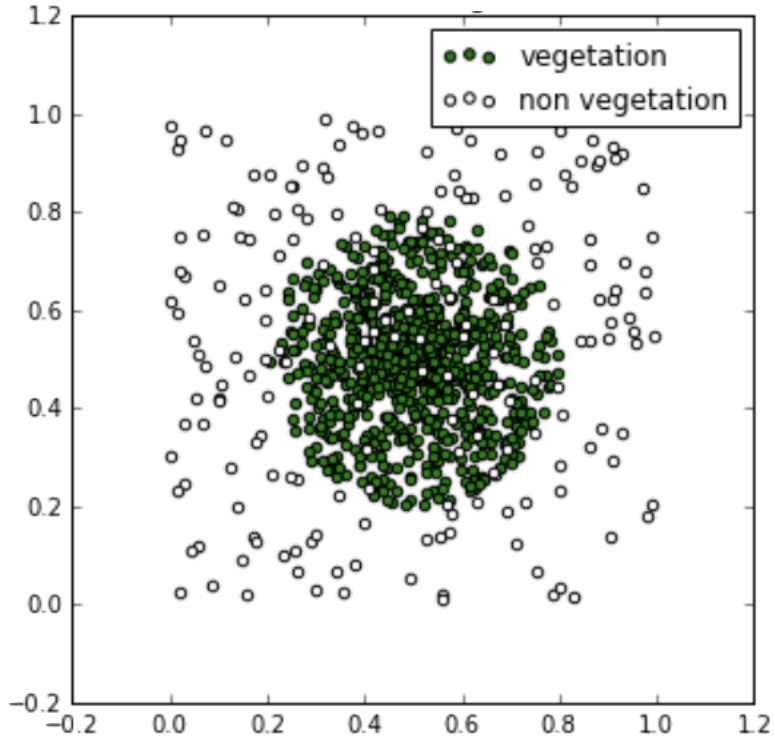


Week 6
Geometry of Data

University of Salford MANCHESTER

- **Question:** How about these?

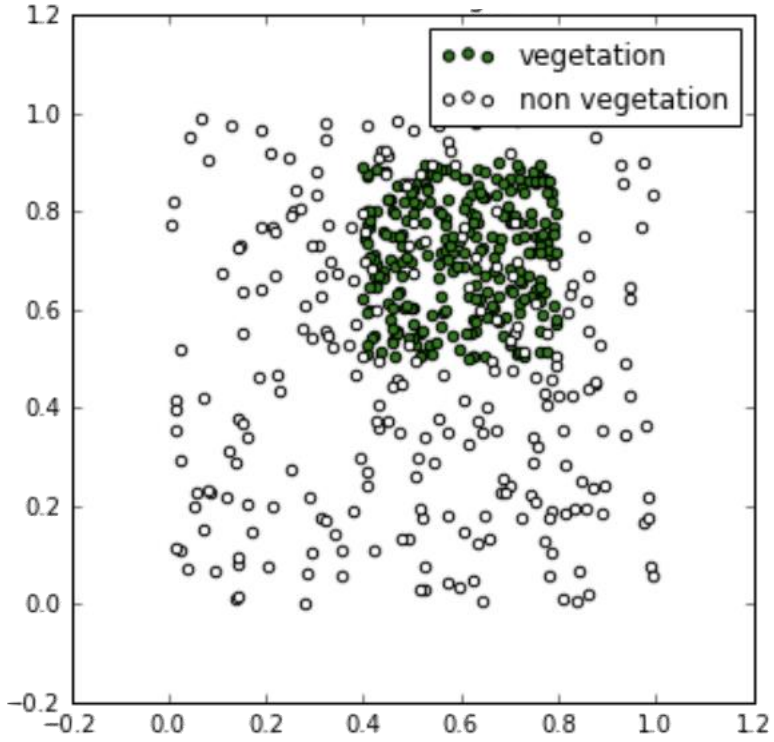


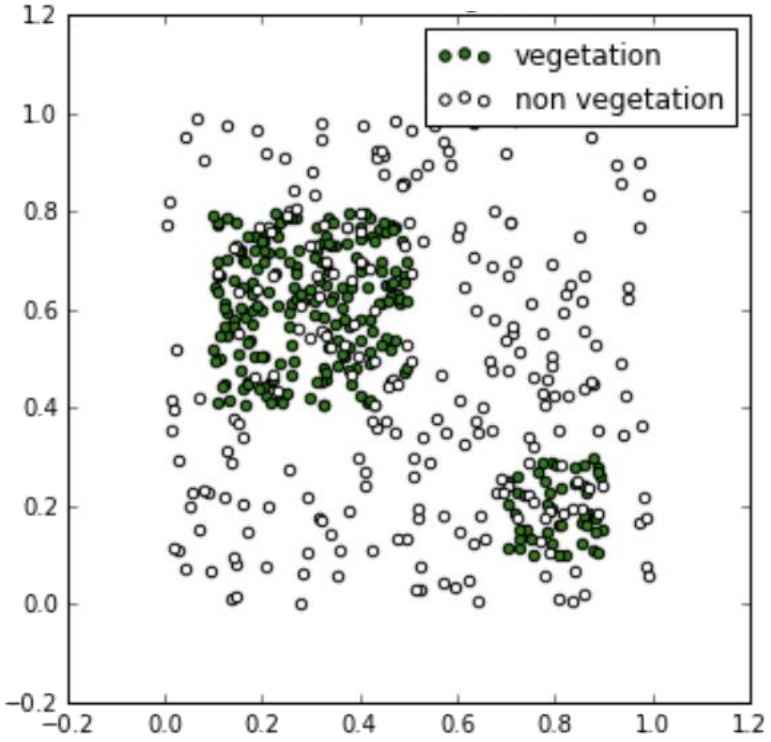


Week 6
Geometry of Data

University of Salford MANCHESTER

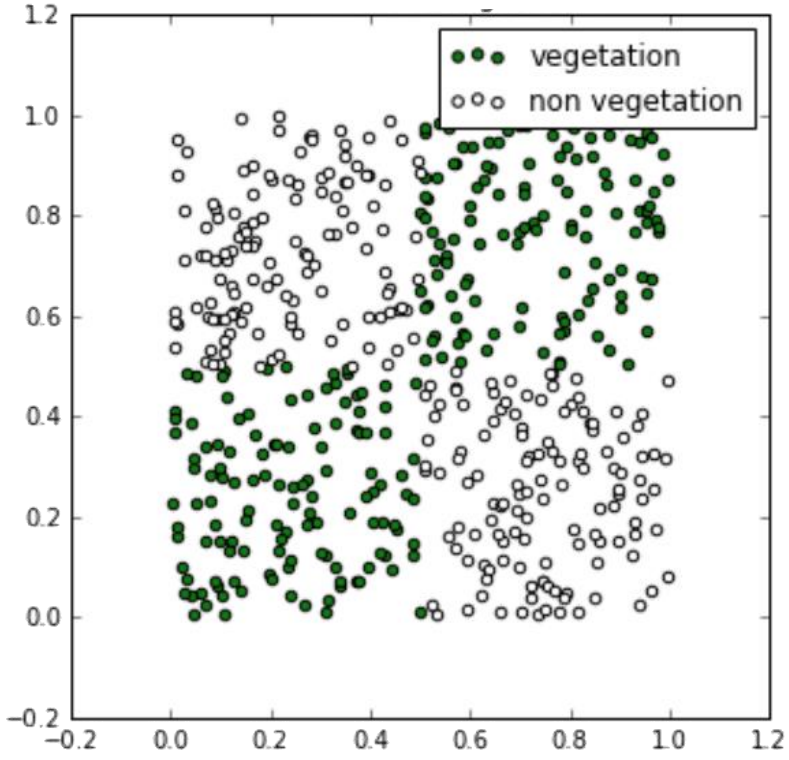
- **Question:** Or these?



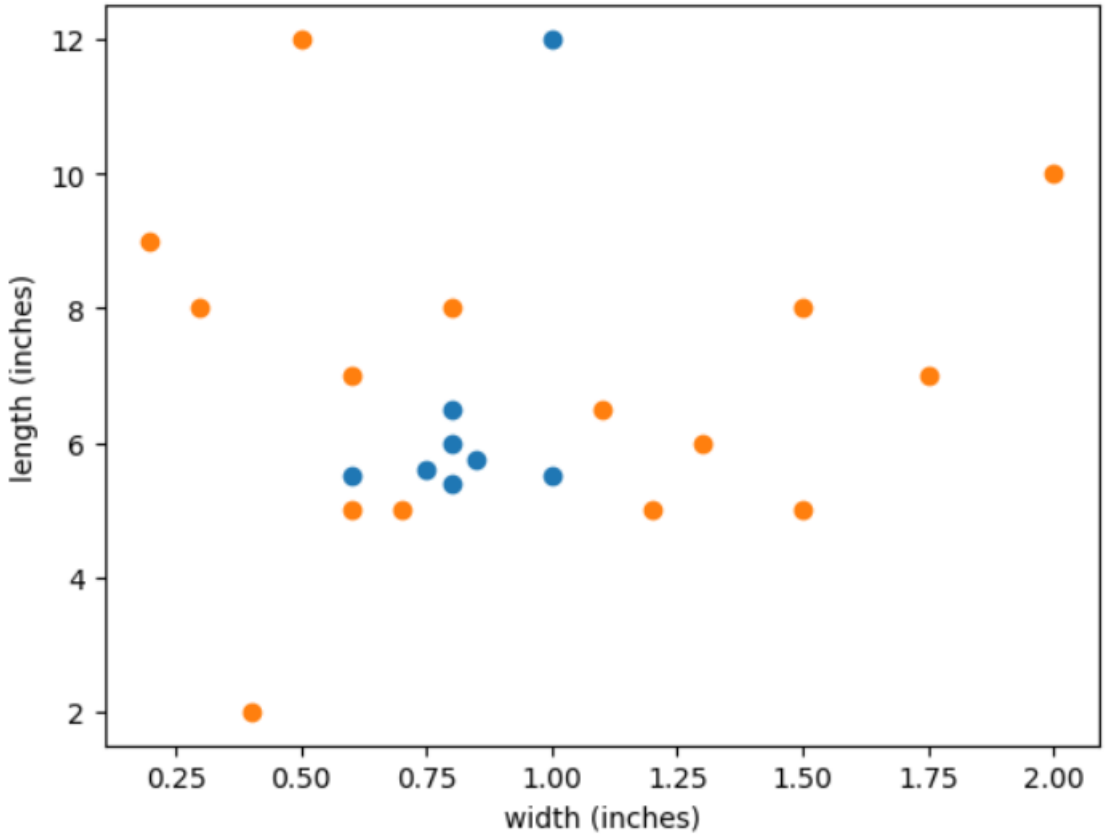


Week 6
Geometry of Data

•Notice that in all of the datasets the classes are still well-separated in the feature space, but ***the decision boundaries cannot easily be described by single equations***:



Week 6
Geometry of Data



University of Salford MANCHESTER

Non Linear Models

- KNN
- Decision Trees

University of Salford MANCHESTER

Non Linear Models

- KNN
  - Regression
  - Classification

Week 6
Geometry of Data

University of Salford MANCHESTER

- While logistic regression models with linear boundaries are intuitive to interpret by examining the impact of each predictor on the log-odds of a positive classification, it is less straightforward to interpret nonlinear decision boundaries in context:

- $(x_3+2x_2) - x_1^2 + 10 = 0$

- It would be desirable to build models that:

1. allow for ***complex decision boundaries***.

Week 6 – More ML
K - Nearest Neighbours

University of Salford MANCHESTER

- Not every ML method builds a model!
- Main idea: Uses the similarity between examples.
- Assumption: Two similar examples should have same labels.
- Assumes all examples (instances) are points in the d dimensional space

University of Salford MANCHESTER

k-NN for Regression

Week 6 – More ML
K - Nearest Neighbours

University of Salford MANCHESTER

The ***k-Nearest Neighbor (kNN) model*** is an intuitive way to predict a quantitative response variable:

> *to predict a response for a set of observed predictor values, we use the responses of other observations most similar to it*

Note: this strategy can also be applied in classification to predict a categorical variable. We will encounter kNN again later in the course in the context of classification.

Week 6 – More ML
K - Nearest Neighbours

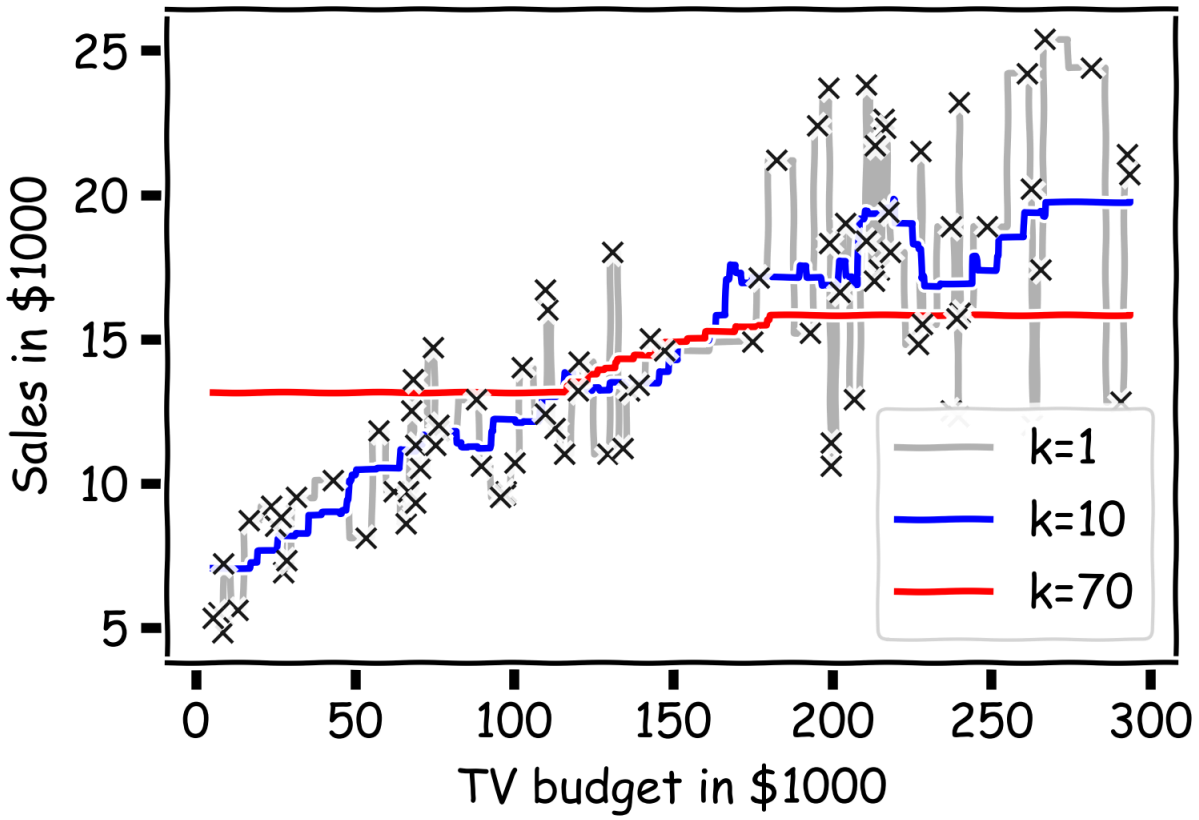
University of Salford MANCHESTER

For a fixed a value of k, the predicted response for the i-th observation is the average of the observed response of the k-closest observations:

$$\hat{y}_n = \frac{1}{k} \sum_{i=1}^{k} y_{n_i}$$

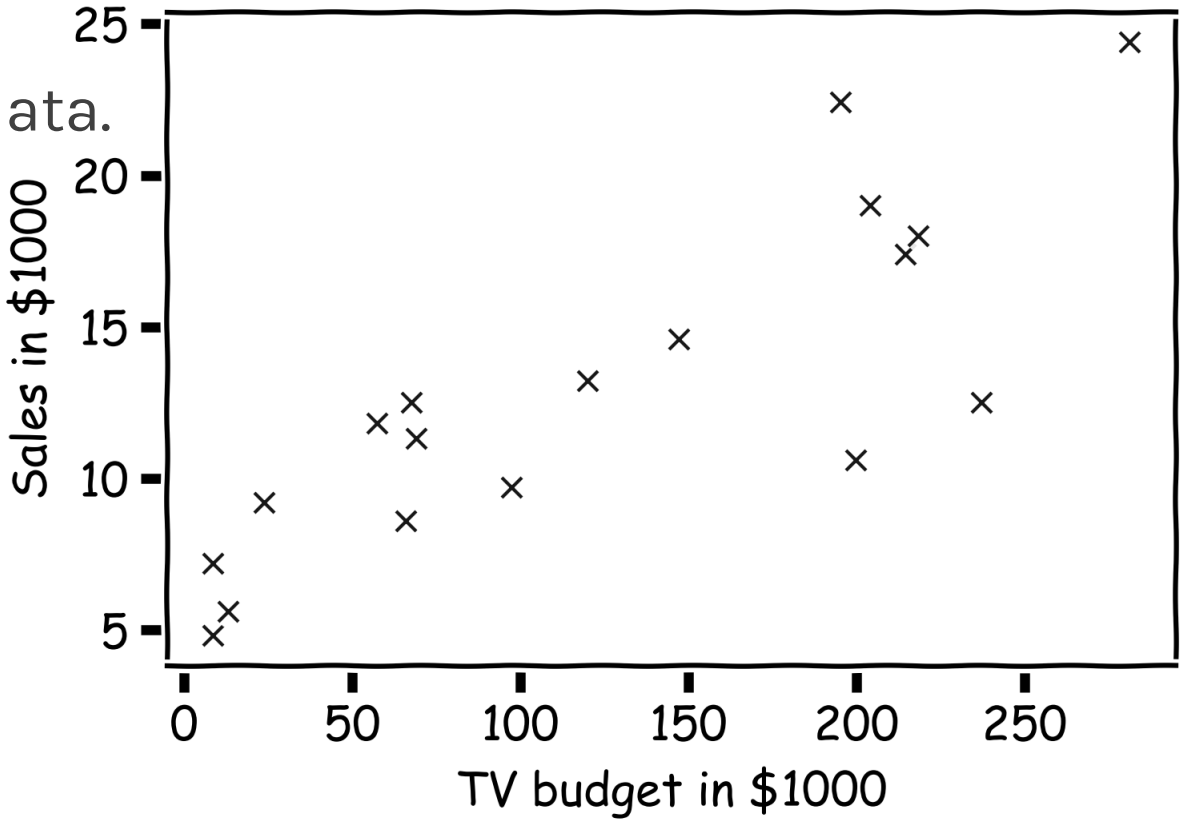
where $\{x_{n1}, \dots, x_{nk}\}$ are the k observations most similar to x_i (*similar* refers to a notion of distance between predictors).

Week 6 – More ML
K - Nearest Neighbours



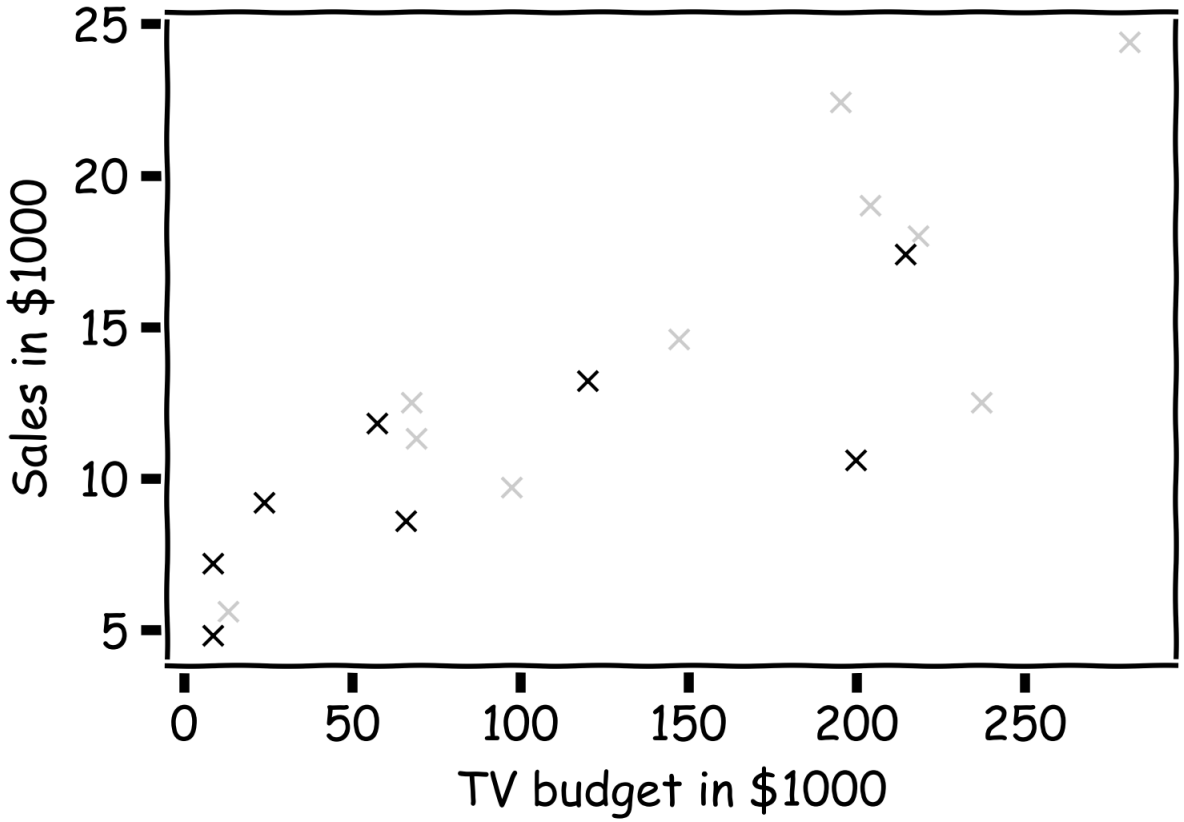
Week 6 – More ML
KNN - Error Evaluation

Start with some data.



Week 6 – More ML
KNN - Error Evaluation

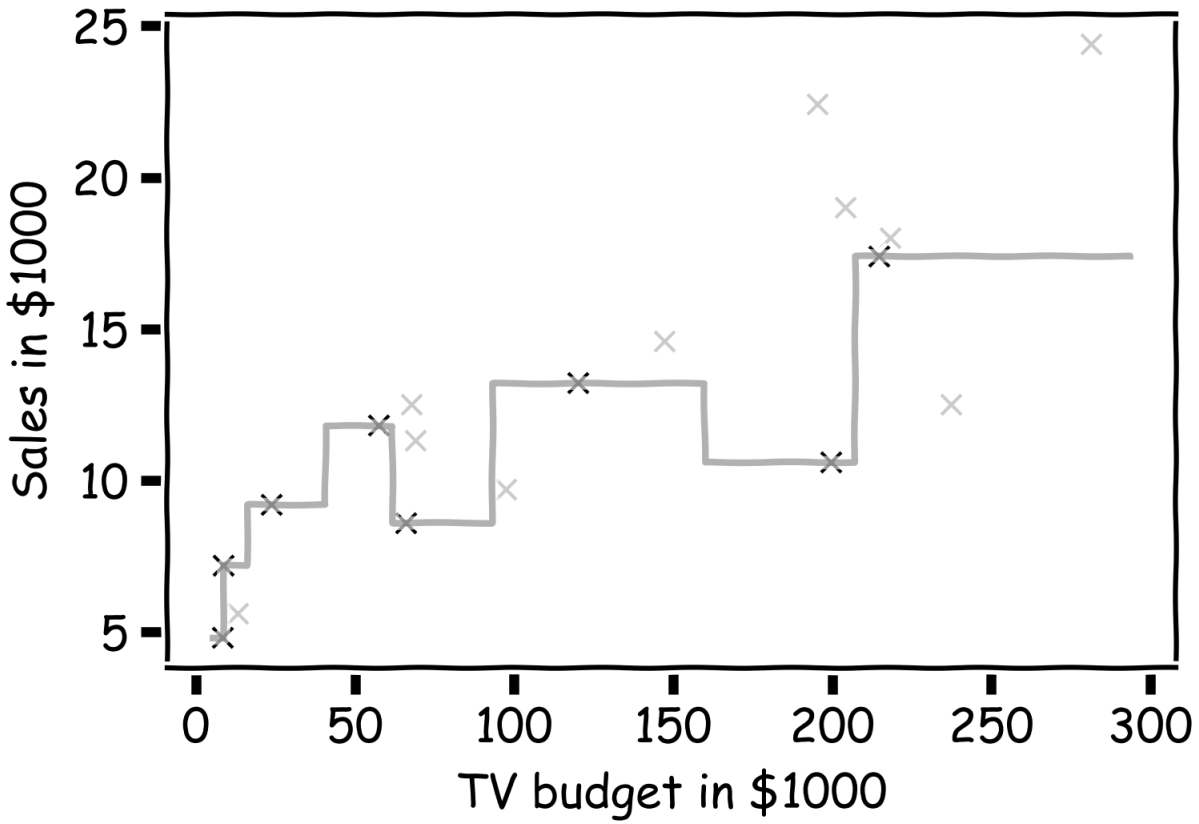
Hide some of the data from the model. This is called **train-test** split.



We use the train set to estimate $\hat{y}$, and the test set to evaluate the model.

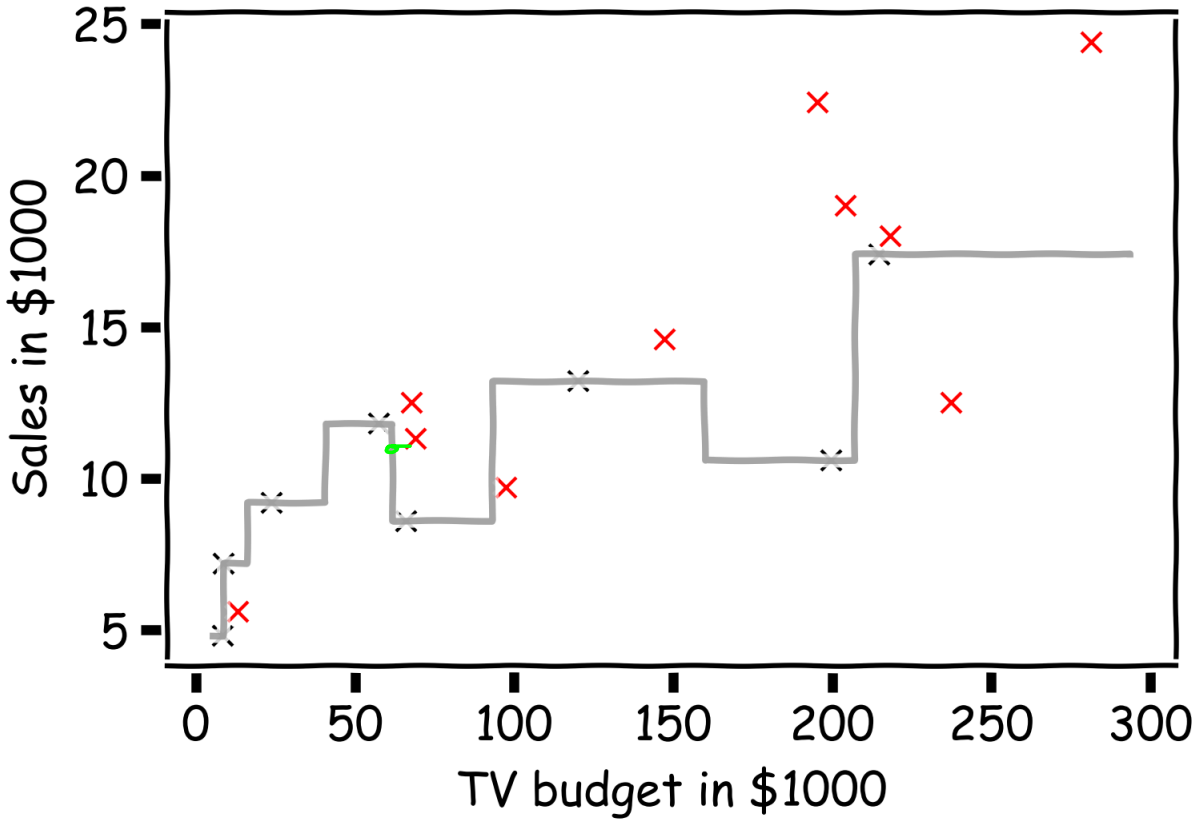
Week 6 – More ML
KNN - Error Evaluation

Estimate $\hat{y}$ for *k=1*.

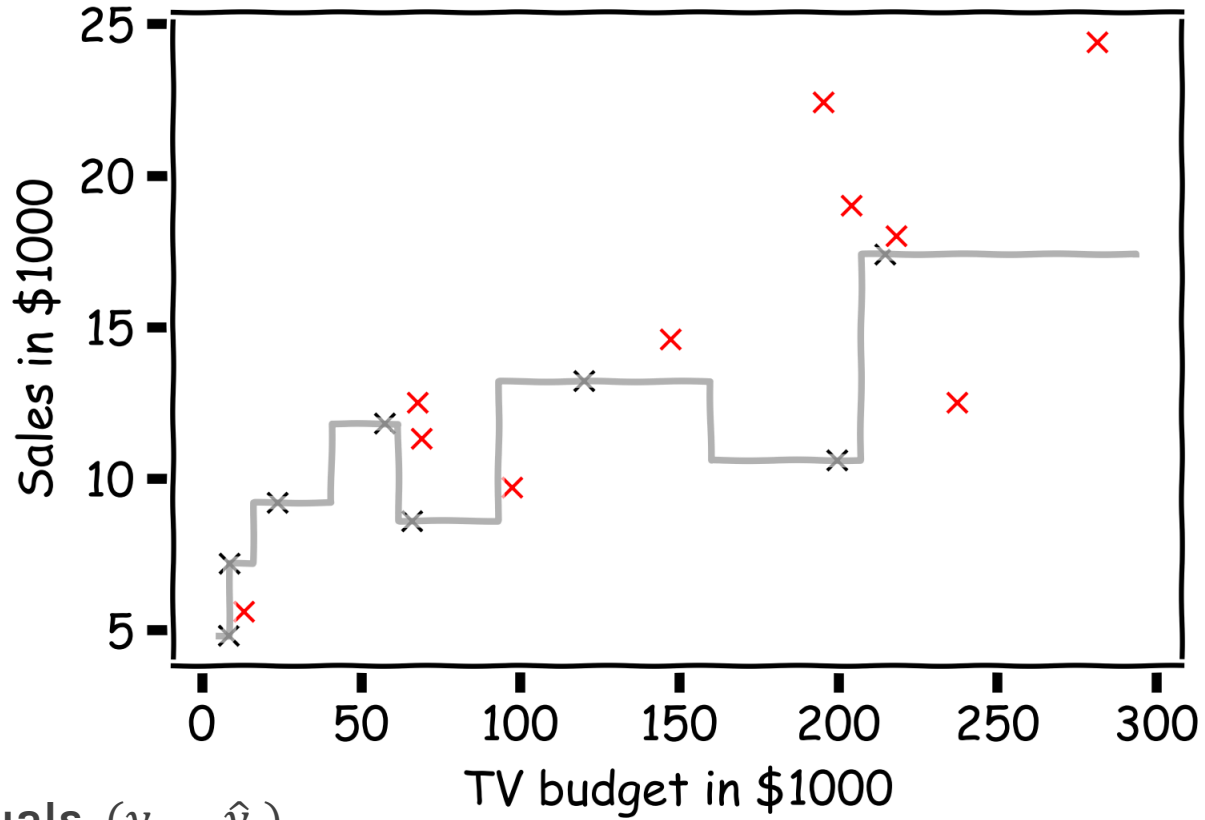


Week 6 – More ML
KNN - Error Evaluation

Now, we look at the data we have not used, the **test data** (red crosses).



Week 6 – More ML
KNN - Error Evaluation



Calculate the **residuals** $(y_i - \hat{y}_i)$.

Week 6 – More ML
KNN - Error Evaluation

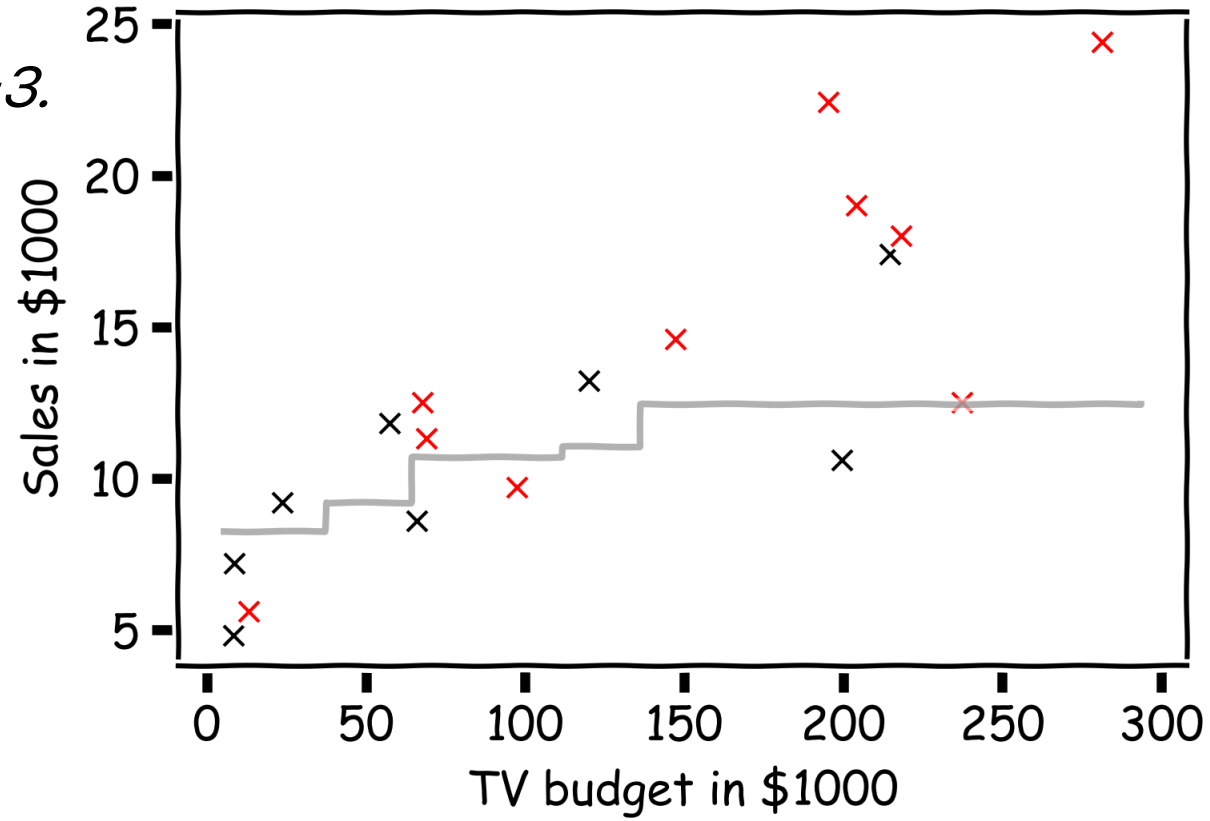
University of Salford MANCHESTER

- KNN uses the standard **Euclidian distance** to define nearest neighbors.
  Given two examples x_i and x_j:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$$

Week 6 – More ML
KNN - Error Evaluation

Do the same for *k=3.*



Week 6 – More ML
K - Nearest Neighbours

University of Salford MANCHESTER

```
from sklearn.neighbors import KNeighborsRegressor
neigh = KNeighborsRegressor(n_neighbors=2)
neigh.fit(X_train, y_train)
```

```
KNeighborsRegressor(n_neighbors=2)
```

```
L_fit = np.arange(X_train.min(), X_train.max(), 1)[:, np.newaxis]
L_fitTest = np.arange(X_test.min(), X_test.max(), 1)[:, np.newaxis]

# linear fit
print("Results for Linear features")
regr = neigh.fit(X_train,y_train)
y_lin_fit = neigh.predict(L_fit)
y_lin_fit_test = neigh.predict(L_fitTest)
neigh_r2 = r2_score(y_train, neigh.predict(X_train))
neigh_r2_Test = r2_score(y_test, neigh.predict(X_test))
print("r^2 on raining dataset",neigh_r2)
print("r^2 on testing dataset",neigh_r2_Test)
```

```
Results for Linear features
r^2 on raining dataset 0.8406182674175111
r^2 on testing dataset 0.41876700247674137
```

Week 6 – More ML
K - Nearest Neighbours

University of Salford MANCHESTER

```
from sklearn.neighbors import KNeighborsRegressor
neigh = KNeighborsRegressor(n_neighbors=9)
neigh.fit(X_train, y_train)
```

```
KNeighborsRegressor(n_neighbors=9)
```

```
L_fit = np.arange(X_train.min(), X_train.max(),
L_fitTest = np.arange(X_test.min(), X_test.max(

# linear fit
print("Results for Linear features")
regr = neigh.fit(X_train,y_train)
y_lin_fit = neigh.predict(L_fit)
y_lin_fit_test = neigh.predict(L_fitTest)
neigh_r2 = r2_score(y_train, neigh.predict(X_tr
neigh_r2_Test = r2_score(y_test, neigh.predict(
print("r^2 on raining dataset",neigh_r2)
print("r^2 on testing dataset",neigh_r2_Test)
```

```
Results for Linear features
r^2 on raining dataset 0.7139300319996937
r^2 on testing dataset 0.6120840957458344
```

University of Salford MANCHESTER

k-NN for Classification

Week 6 – More ML
KNN - Classification

University of Salford MANCHESTER

- How can we modify the *k*-NN approach for classification?
- The approach here is the same as for *k*-NN regression: use the other available observations that are most similar to the observation we are trying to predict (classify into a group) based on the predictors at hand.
- How do we classify which category a specific observation should be in based on its nearest neighbors?
- The category that shows up the most among the nearest neighbors.

Week 6 – More ML
KNN - Classification

- The *k*-NN classifier first identifies the *k* points in the training data that are closest to x_0, represented by $\mathcal{N}_0$. It then estimates the conditional probability for class *j* as the fraction of points in $\mathcal{N}_0$ whose response values equal *j*:

$$P(Y = j|X = x_0) = \frac{1}{k}\sum_{i\in\mathcal{N}_0} I(y_i = j)$$

- Then, the *k*-NN classifier applies Bayes rule and classifies the test observation, x_0, to the class with largest estimated probability.

Week 6 – More ML
K – NN (Binary classifier)

University of Salford MANCHESTER

Training algorithm:

Add each training example (x, y) to the dataset $\mathcal{D}$.

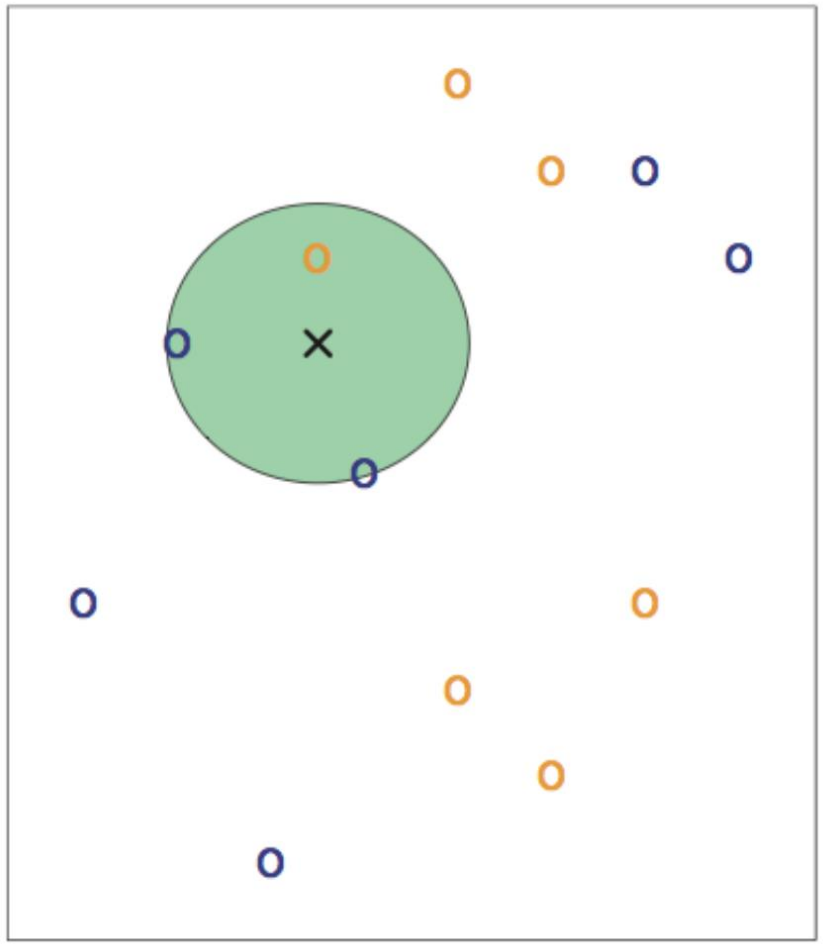
$x \in \mathbb{R}^d$, $y \in \{+1, -1\}$.

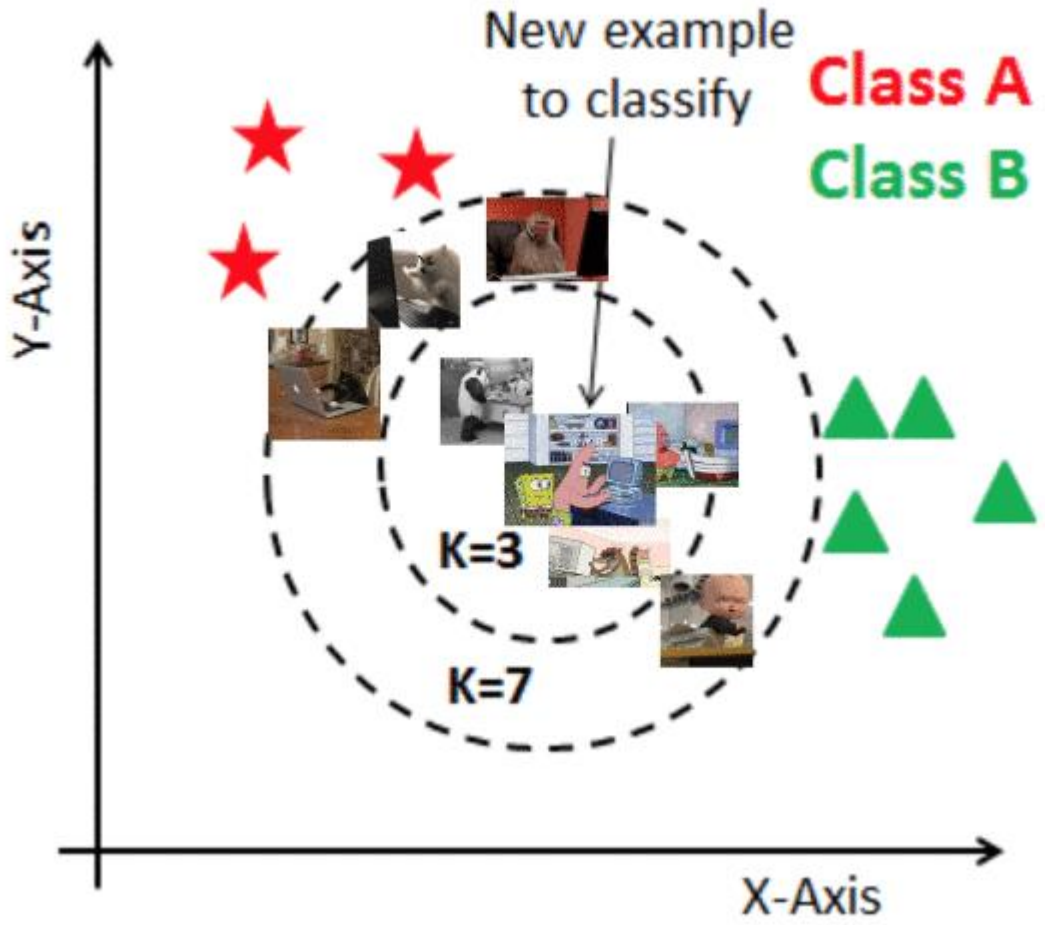
Classification algorithm:

Given an example x_q to be classified. Suppose $N_k(x_q)$ is the set of the K-nearest neighbors of x_q.

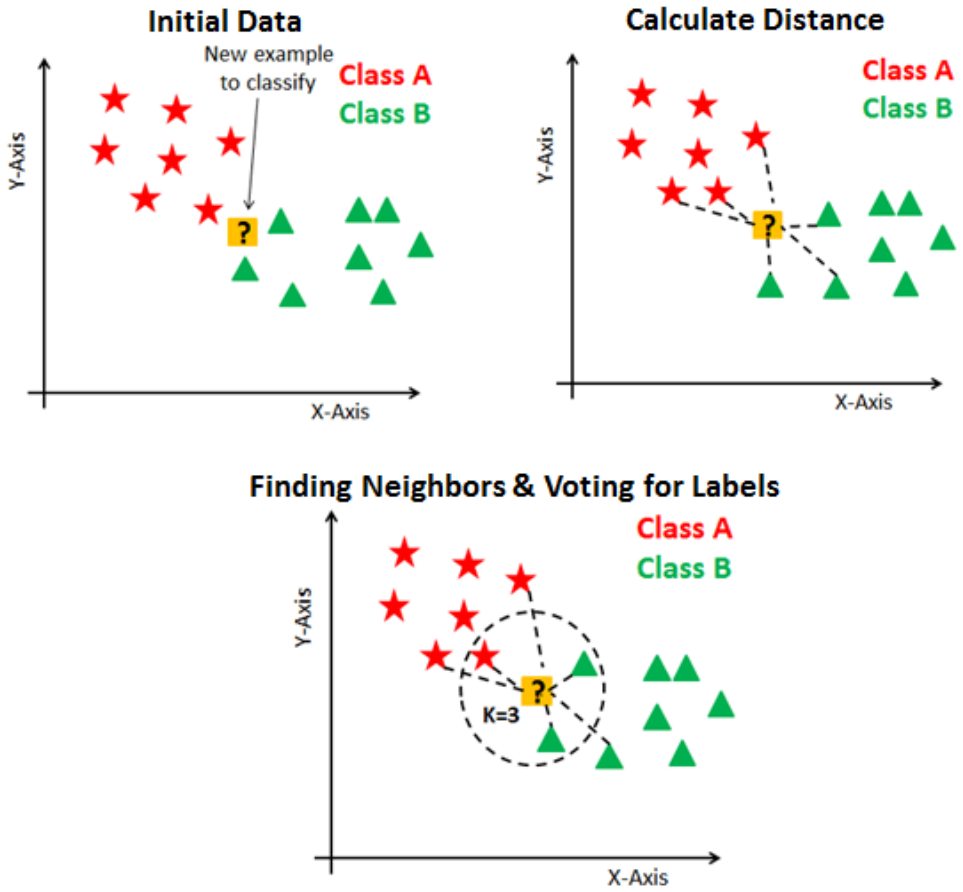
$$\hat{y}_q = sign\Big(\sum_{x_i \in N_k(x_q)} y_i\Big)$$

Week 6 – More ML
K - Nearest Neighbours





Week 6 – More ML
K - Nearest Neighbours



Week 6 – More ML
KNN - Classification

University of Salford MANCHESTER

```
clf_neigh = KNeighborsClassifier(n_neighbors=5)
#Training
clf_neigh.fit(X_train, y_train)
# Make a prediction
y_pred = clf_neigh.predict(X_test)
print ("The accuracy on validation dataset of KNN: \t", metrics.accuracy_score(y_test, y_pred))
```

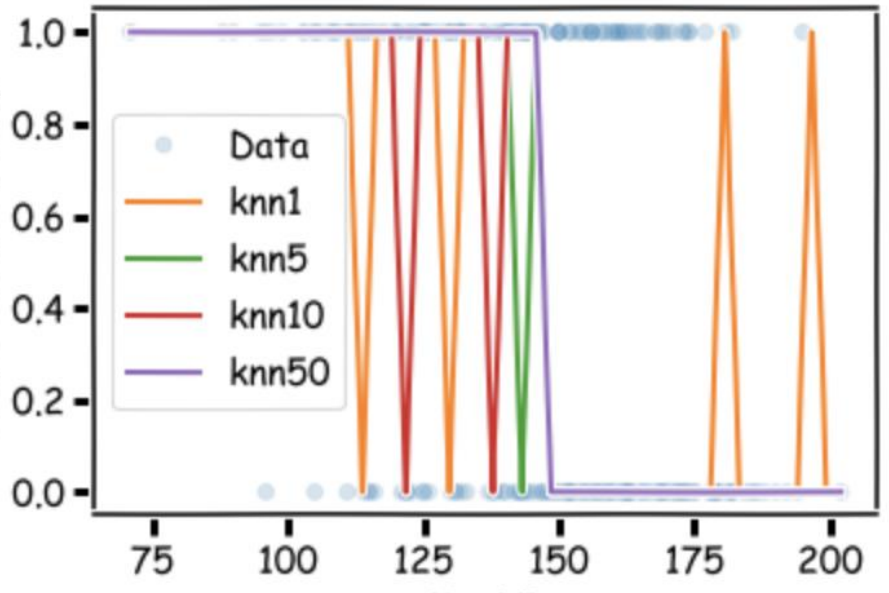
```
The accuracy on validation dataset of KNN:       0.4742268041237113
```

Week 6 - Classification Boundaries in *k*-NN Classification

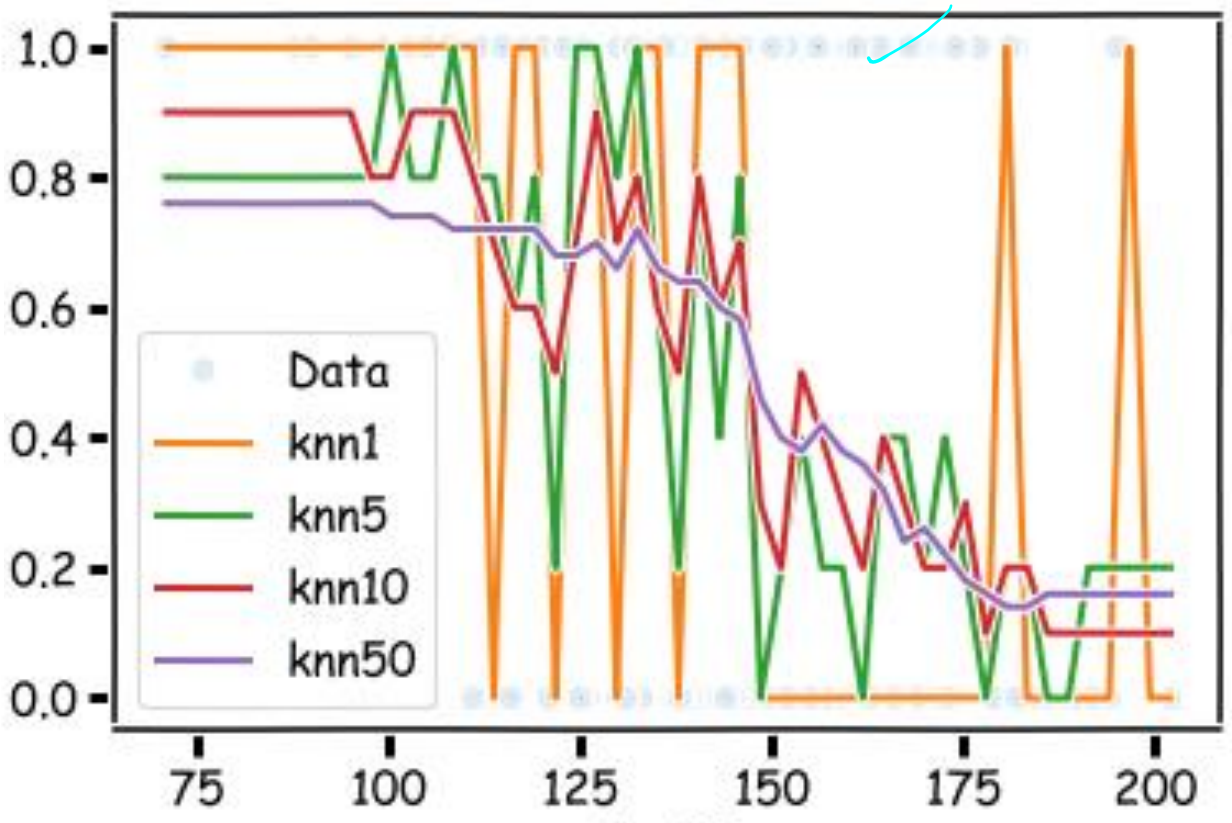
University of Salford MANCHESTER

- What will the classification boundaries look line in k-NN classification? With one predictors? With 2 predictors? With 3+ predictors?

- How do they compare to Logistic Regression classification boundaries?



Week 6 – More ML
KNN - Classification



University of Salford MANCHESTER

Interpretable Models

Week 6
Geometry of Data

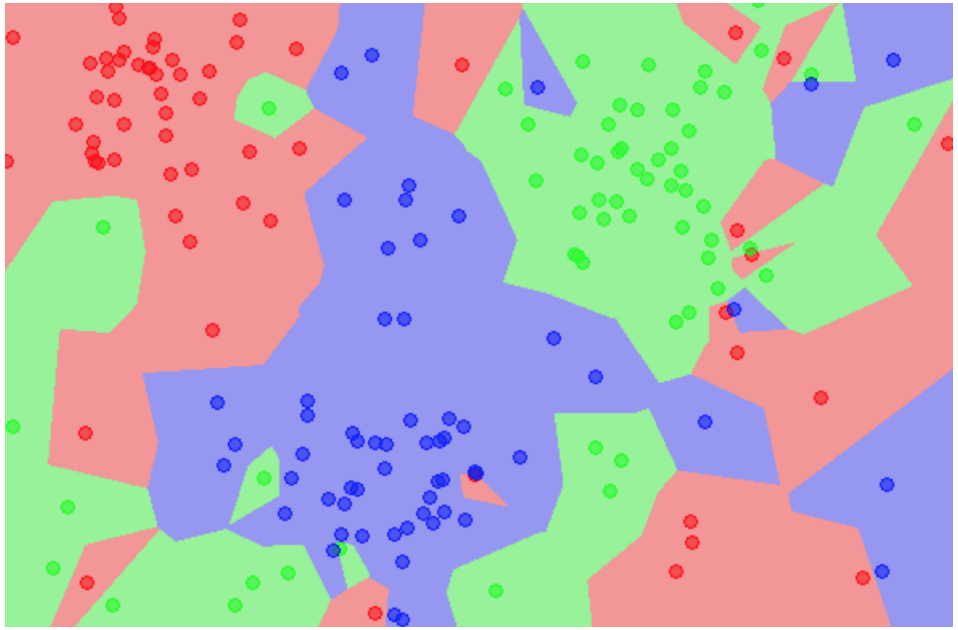
University of Salford MANCHESTER

- While logistic regression models with linear boundaries are intuitive to interpret by examining the impact of each predictor on the log-odds of a positive classification, it is less straightforward to interpret nonlinear decision boundaries in context:

- KNN

- It would be desirable to build models that:

1. allow for ***complex decision boundaries***.
2. are also ***easy to interpret***.
3. fast at development time



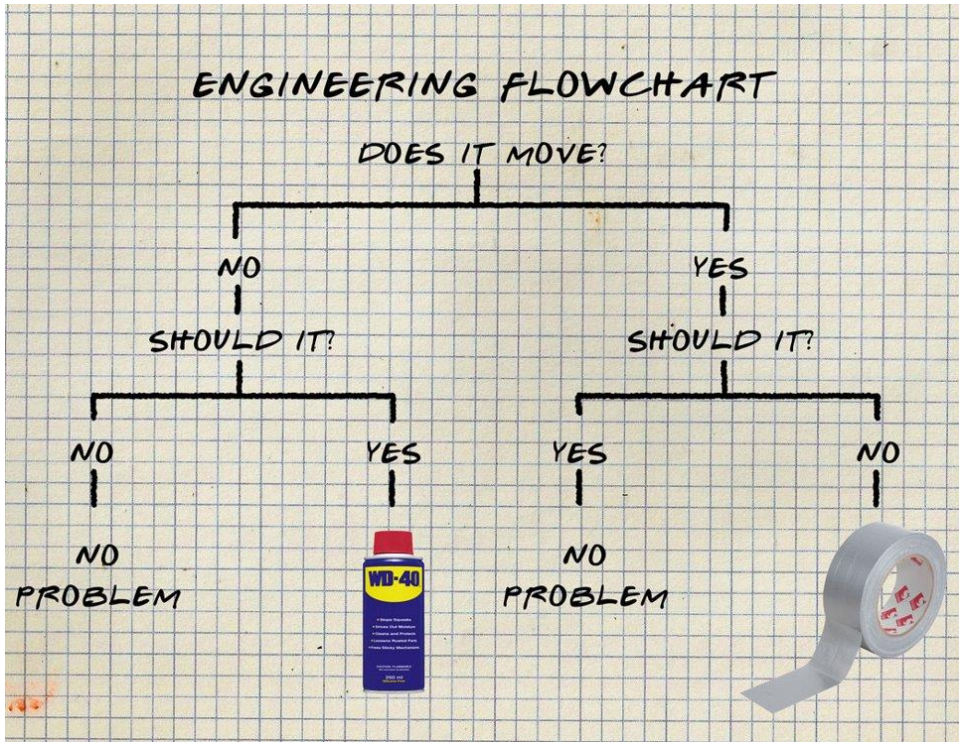
University of Salford MANCHESTER

Non Linear Models

- Decision Trees
  - Regression
  - Classification

Week 6
Interpretable Models

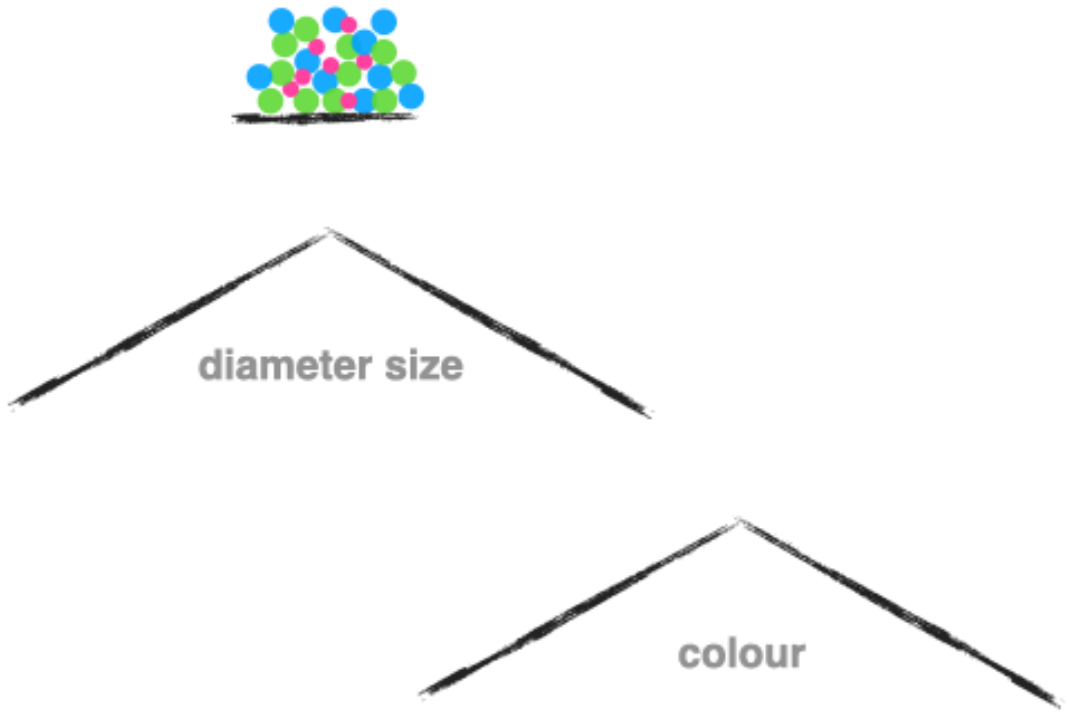
- People in every walk of life have long been using interpretable models for differentiating between classes of objects and phenomena:



Week 6
Interpretable Models

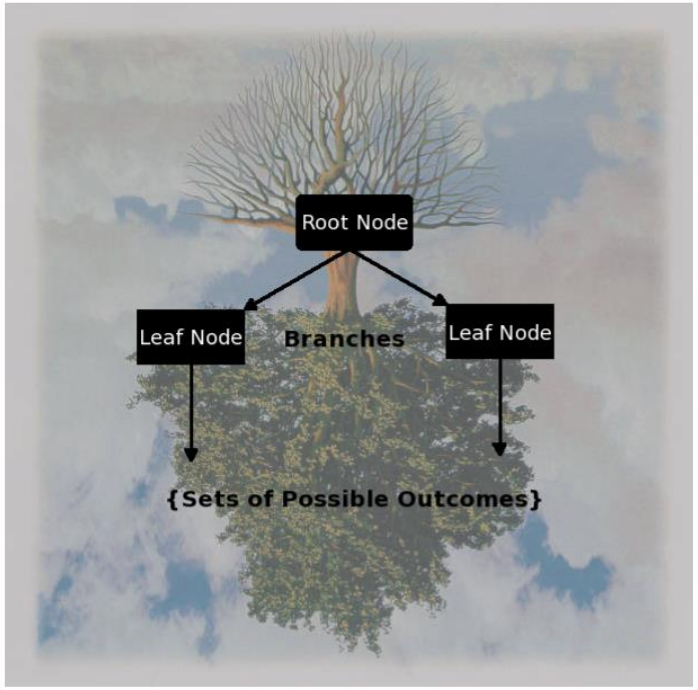
University of Salford MANCHESTER

- People in every walk of life have long been using interpretable models for differentiating between classes of objects and phenomena:



Week 6
Interpretable Models

- People in every walk of life have long been using interpretable models for differentiating between classes of objects and phenomena:



Week 6
Decision Trees

University of Salford MANCHESTER

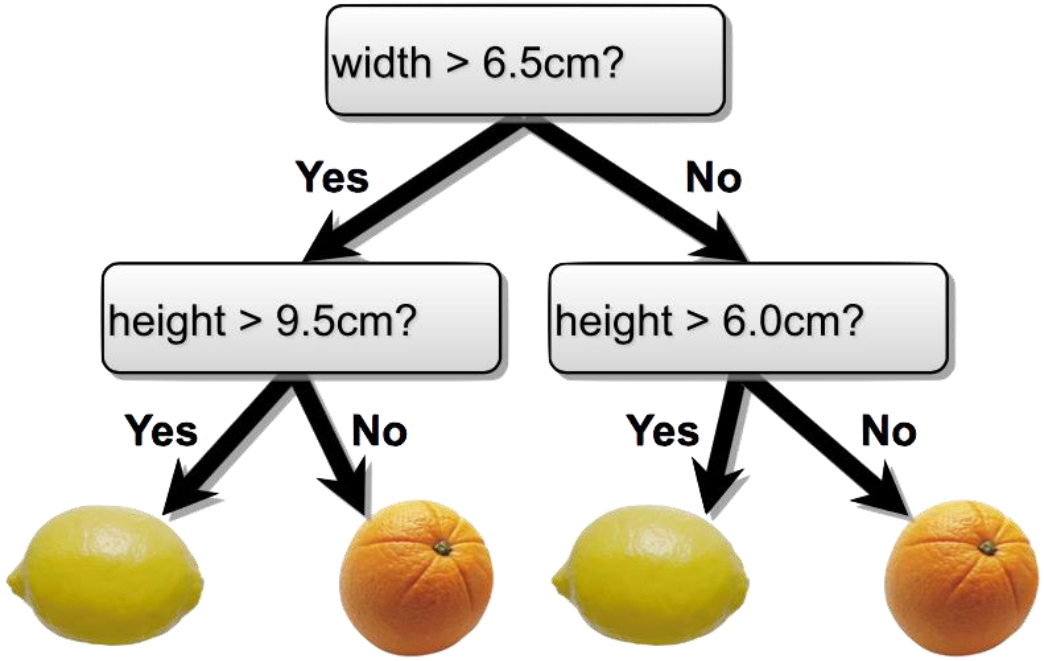
- It turns out that the simple flow charts in our examples can be formulated as mathematical models for classification and these models have the properties we desire; they are:

1. interpretable by humans
2. have sufficiently complex decision boundaries
3. the decision boundaries are locally linear, each component of the decision boundary is simple to describe mathematically.

Week 6
The Geometry of Flow Charts

- Flow charts whose graph is a tree (connected and no cycles) represents a model called a ***decision tree***.

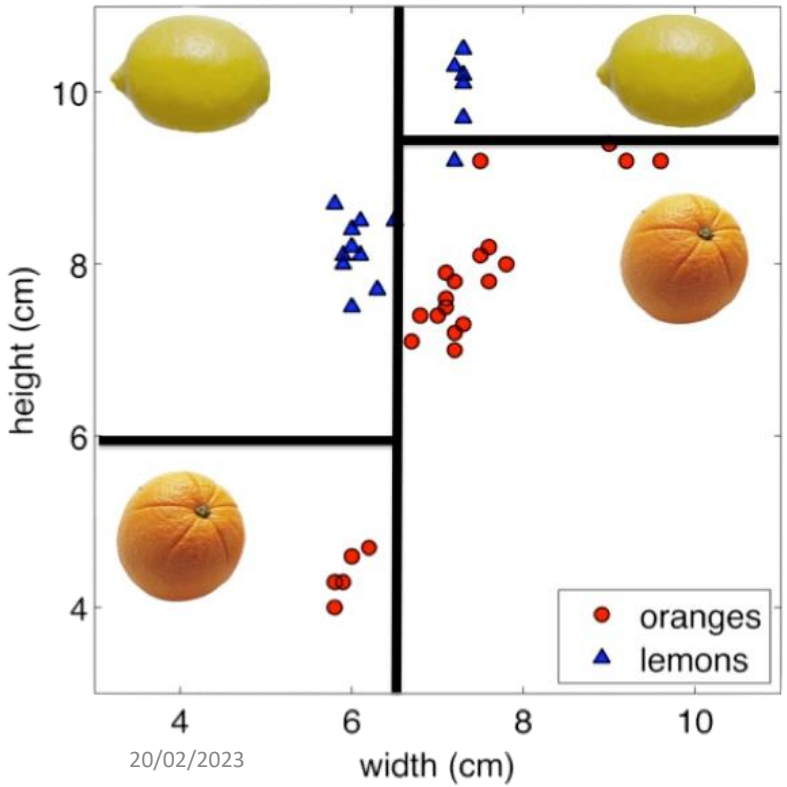
- Formally, a ***decision tree model*** is one in which the final outcome of the model is based on a series of comparisons of the values of predictors against threshold values.

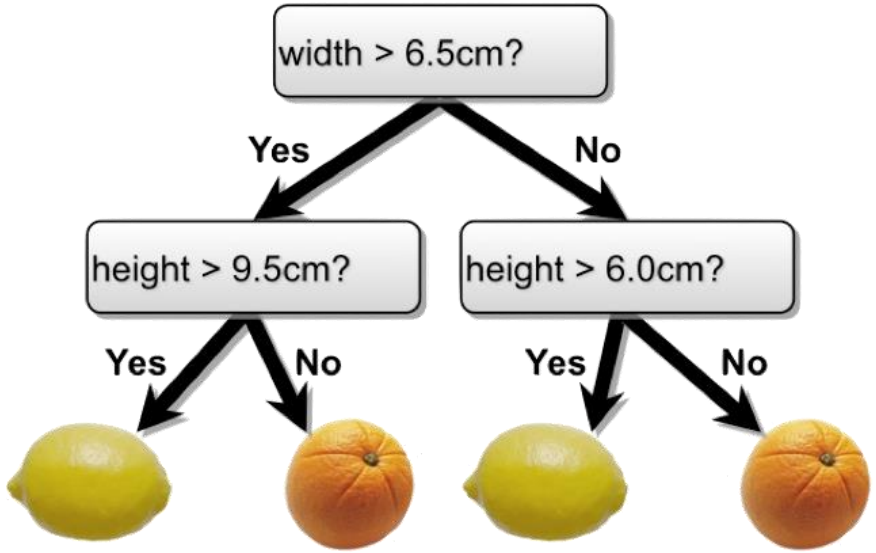


Week 6
The Geometry of Flow Charts

University of Salford MANCHESTER

- Every flow chart tree corresponds to a partition of the feature space by **axis aligned lines or (hyper) planes**. Conversely, every such partition can be written as a flow chart tree.





Width (cm)	Height (cm)	Class
6.8	9.8	
8	6	
10	10	
4	4	

Week 6
Learning the Model

University of Salford MANCHESTER

- Learning the smallest ‘optimal’ decision tree for any given set of data is NP complete for numerous simple definitions of ‘optimal’. Instead, we will seek a reasonably model using a greedy algorithm.

  1. Start with an empty decision tree (undivided feature space)
  2. Choose the ‘optimal’ predictor on which to split and choose the ‘optimal’ threshold value for **splitting**.
  3. Recurse on each new node until ***stopping condition*** is met

- Now, we need only define our **splitting criterion** and **stopping condition**.

Lecture - Week 3 – More ML
Tree Classifiers (Decision Tree)

University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

Week 6
Information Theory

University of Salford MANCHESTER

- One way to quantify the strength of a signal in a particular region is to analyze the distribution of classes within the region. We compute the entropy of this distribution.
- For a random variable with a discrete distribution, the entropy is computed by:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Higher entropy means the distribution is uniform-like (flat histogram) and thus values sampled from it are ‘less predictable’ (all possible values are equally probable).
- Lower entropy means the distribution has more defined peaks and valleys and thus values sampled from it are ‘more predictable’ (values around the peaks are more probable).

Week 6
Information Theory(Decision Tree – C4.5)

University of Salford MANCHESTER

$$\text{Entropy}(S) = \sum_{}^{c} -p_i \log_2 p_i$$

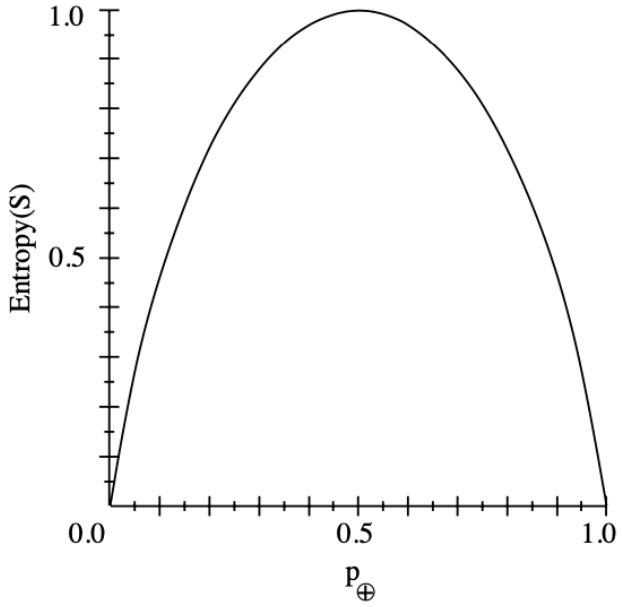
$$\text{Entropy(S)} = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Week 6
Information Theory(Splitting criteria – C4.5)

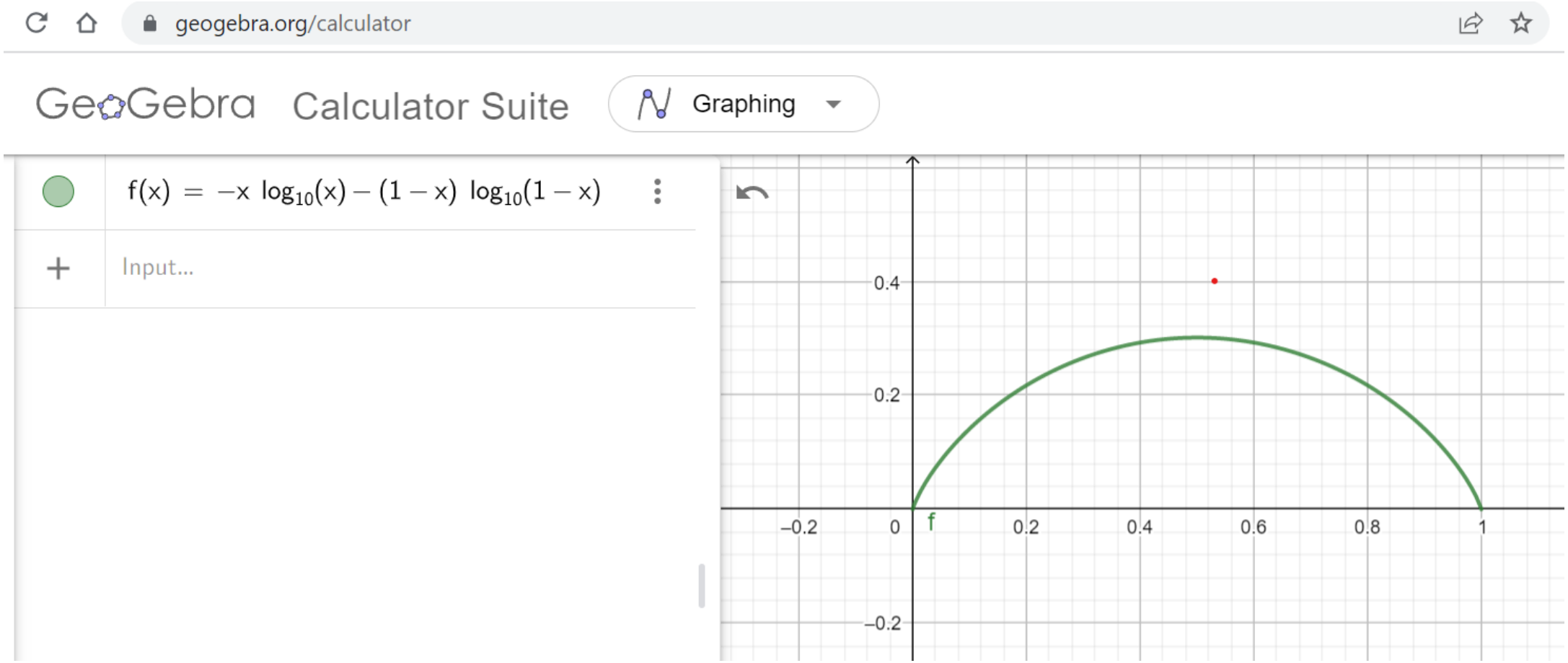
University of Salford MANCHESTER

$$\text{Entropy}(S) = - \; p_{\oplus} \log_2 p_{\oplus} \; - \; p_{\ominus} \log_2 p_{\ominus}$$



Demo

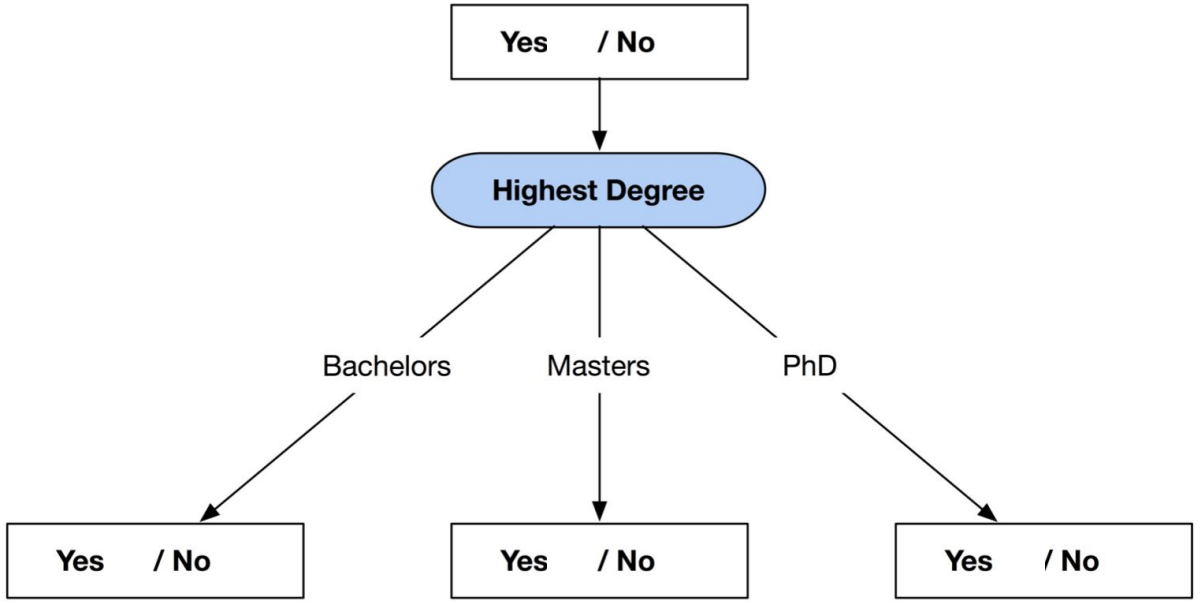
Week 6
Information Theory(Splitting criteria – C4.5)



Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

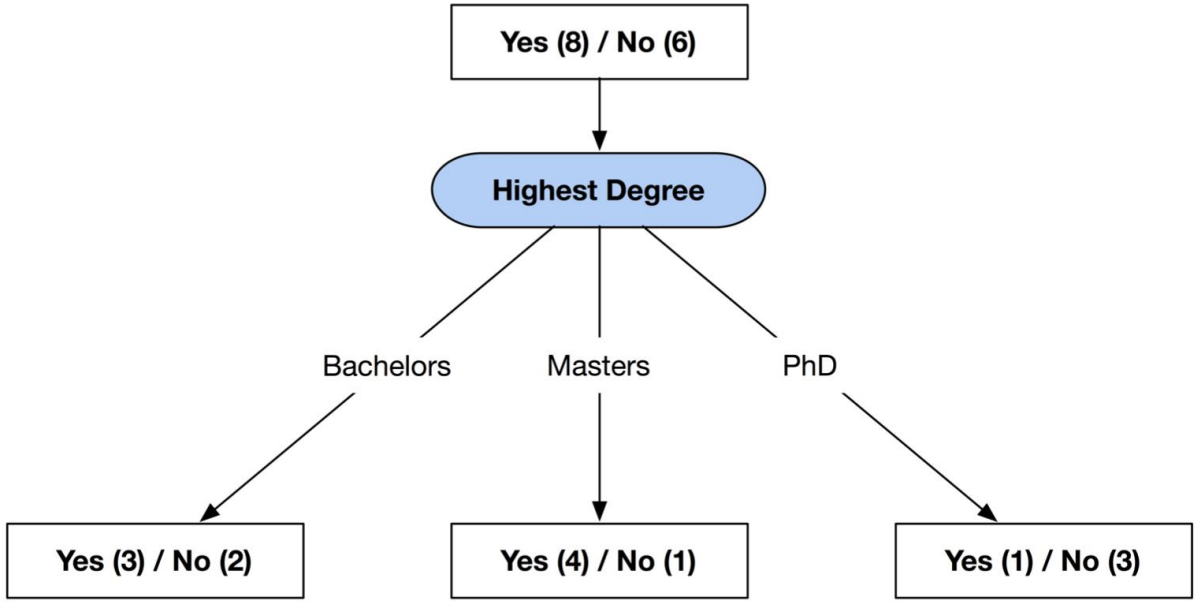
Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?



Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?



Week 6
Information Theory (Decision Tree – C4.5)

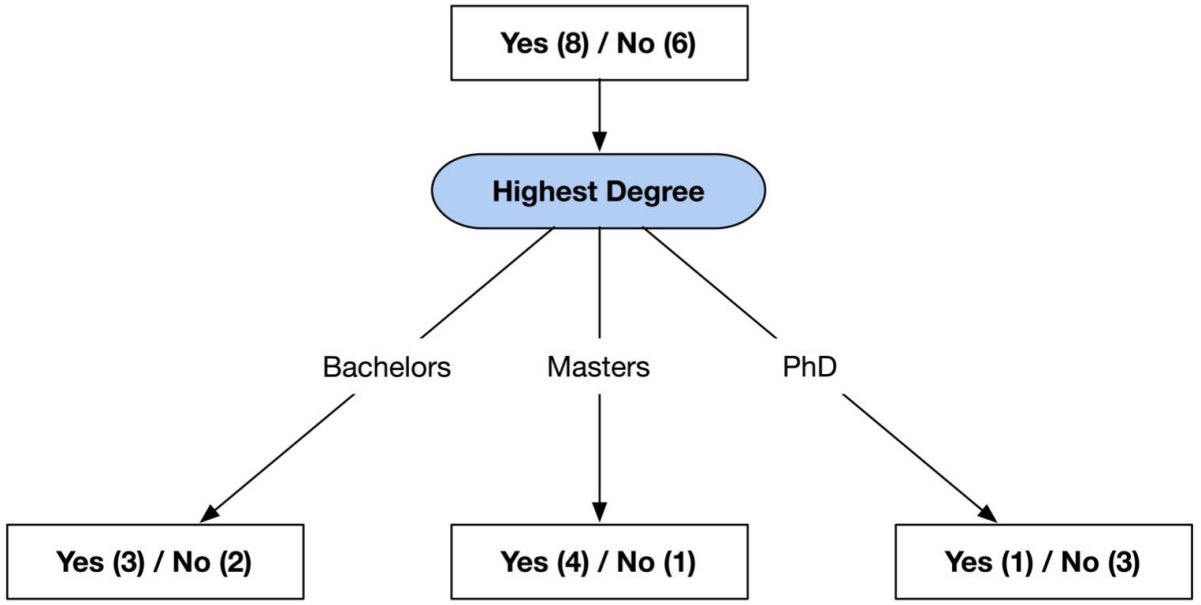
University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

Entropy(S) = - $p_{\oplus}$ $\log_2$ $p_{\oplus}$ − $p_{\ominus}$ $\log_2$ $p_{\ominus}$

Entropy(8, 6) =



Week 6
Information Theory (Decision Tree – C4.5)

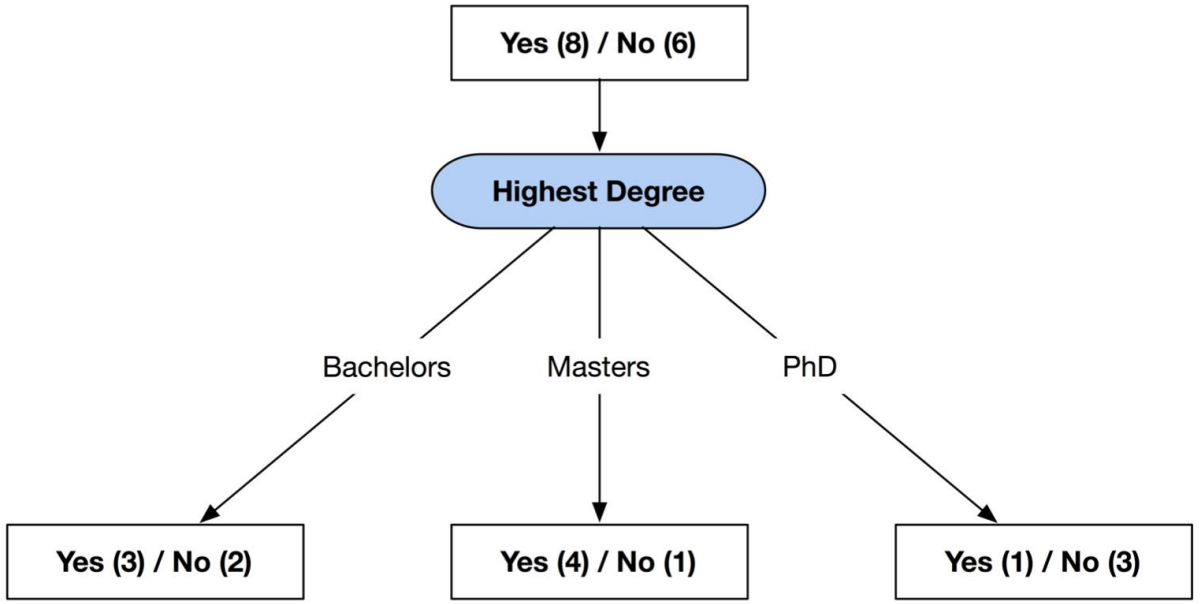
University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$\text{Entropy}(S) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



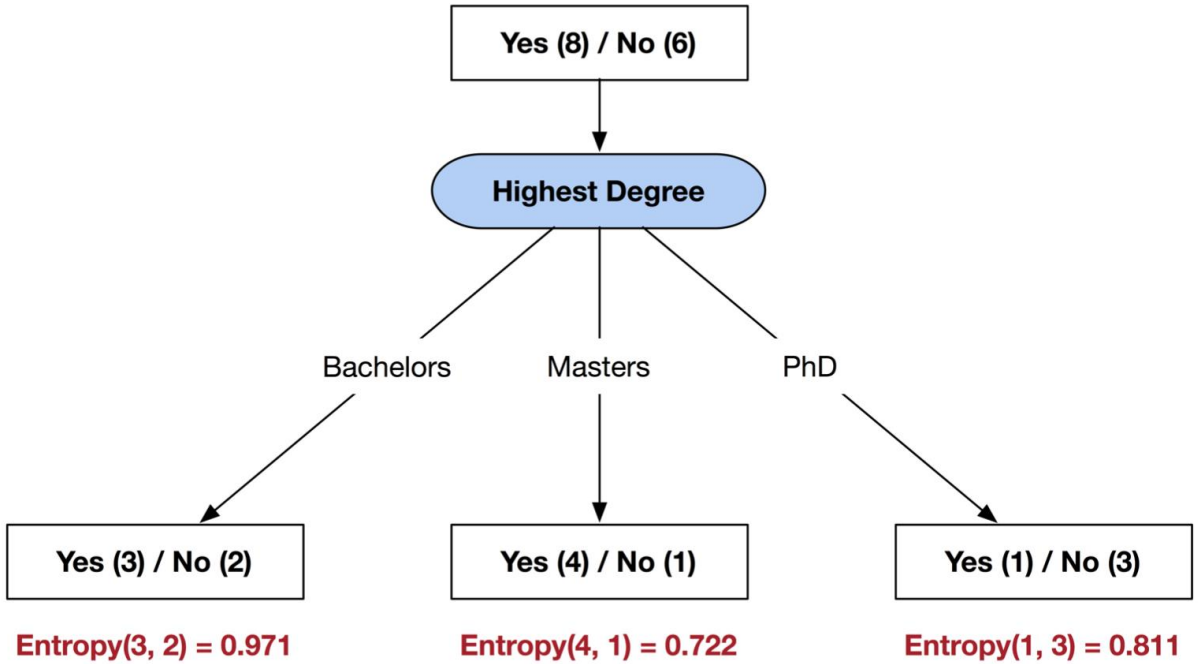
Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$\text{Entropy}(S) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



Week 6
Information Theory (Decision Tree – C4.5)

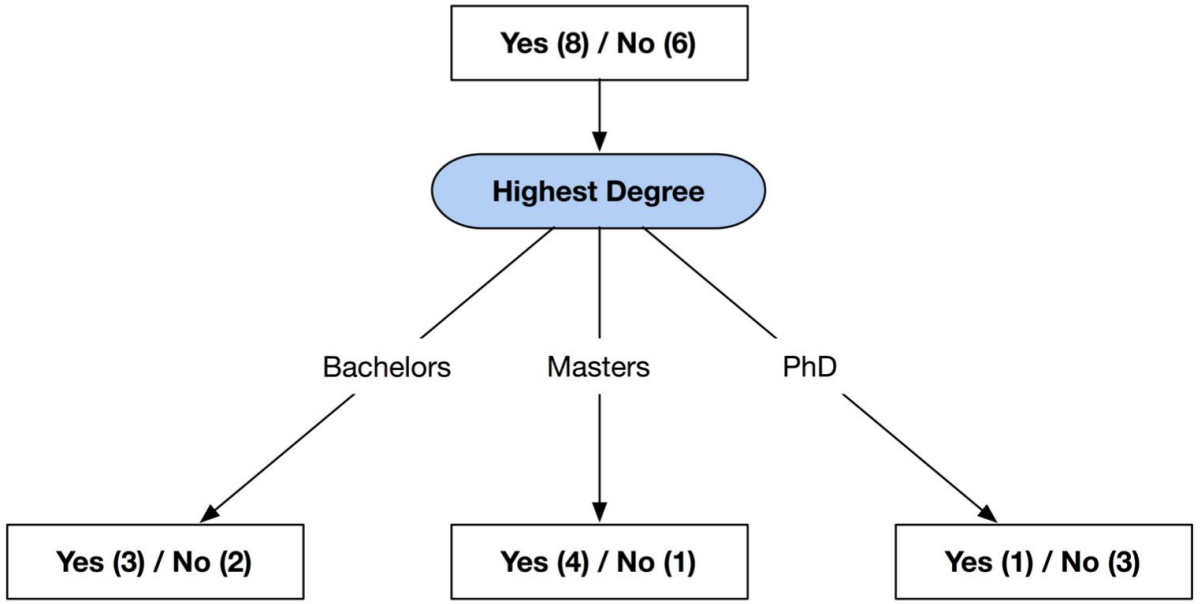
University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



Entropy(3, 2) = 0.971

Entropy(4, 1) = 0.722

Entropy(1, 3) = 0.811

Gain(S, Highest Degree) =

Week 6
Information Theory (Decision Tree – C4.5)

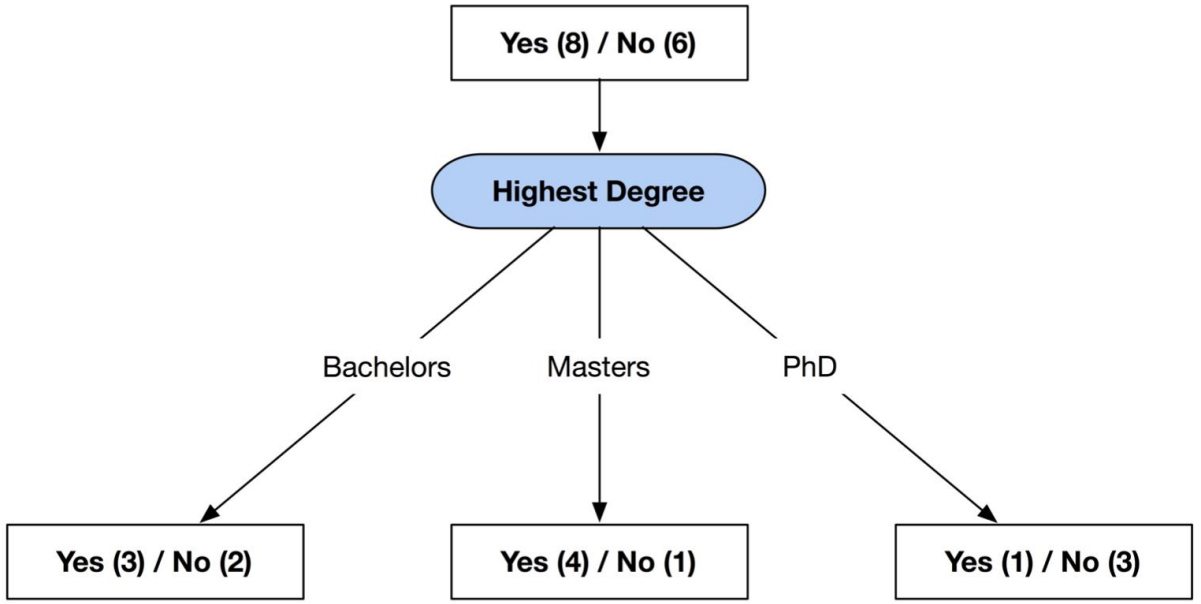
University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



Entropy(3, 2) = 0.971

Entropy(4, 1) = 0.722

Entropy(1, 3) = 0.811

Gain(S, Highest Degree) = 0.985 – (5/14) x 0.971 – (5/14) x 0.722 – (4/14) x 0.811 = 0.149

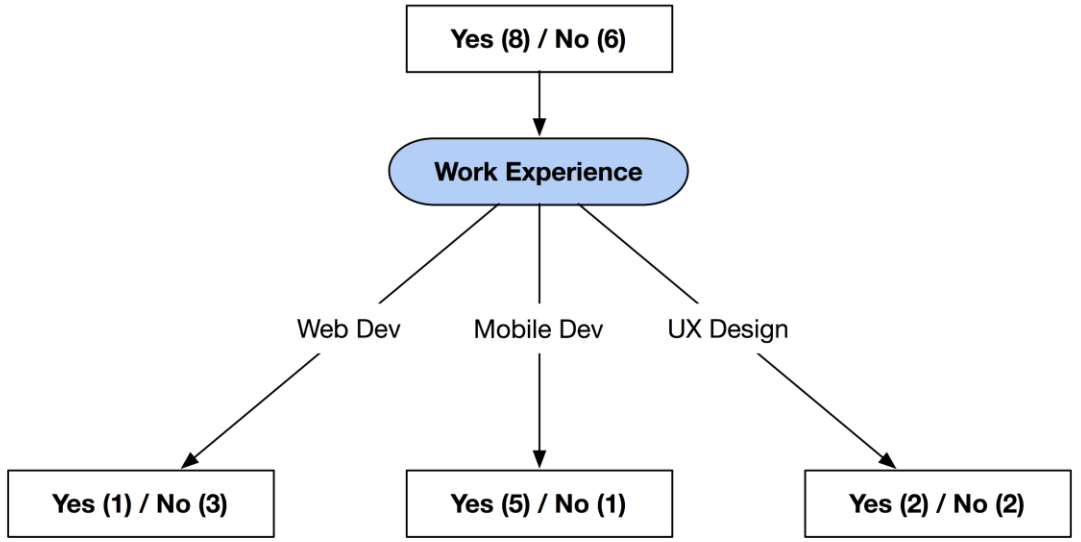
Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) =



Entropy(1, 3) =

Entropy(5, 1) =

Entropy(2, 2) =

Gain(S, Work Experience)

Week 6
Information Theory (Decision Tree – C4.5)

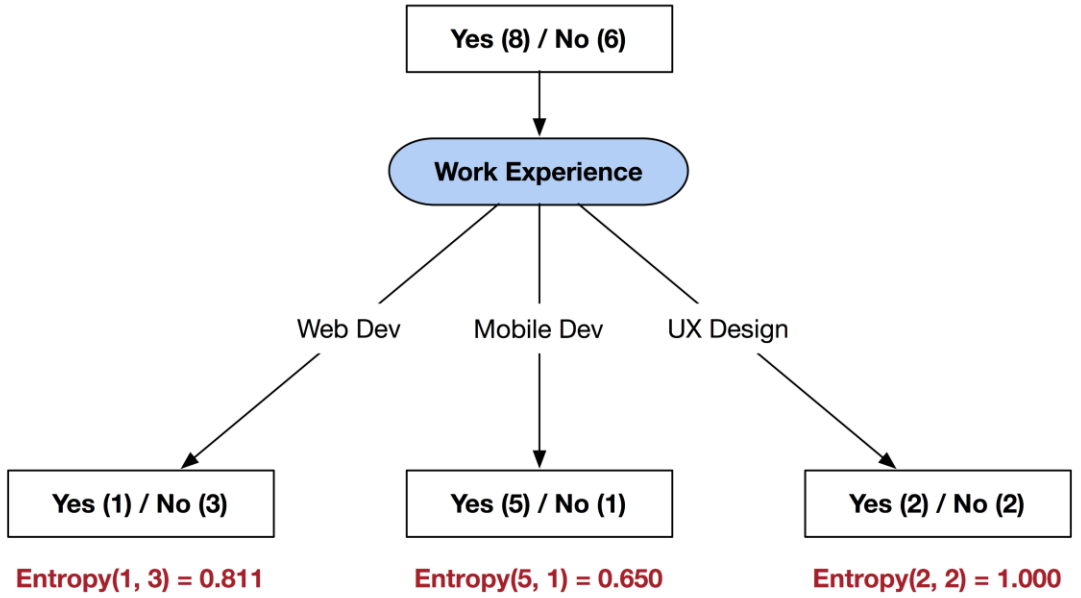
University of Salford MANCHESTER

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



Entropy(1, 3) = 0.811

Entropy(5, 1) = 0.650

Entropy(2, 2) = 1.000

Gain(S, Work Experience) = 0.985 – (4/14) x 0.811 – (6/14) x 0.650 – (4/14) x 1.000 = 0.189

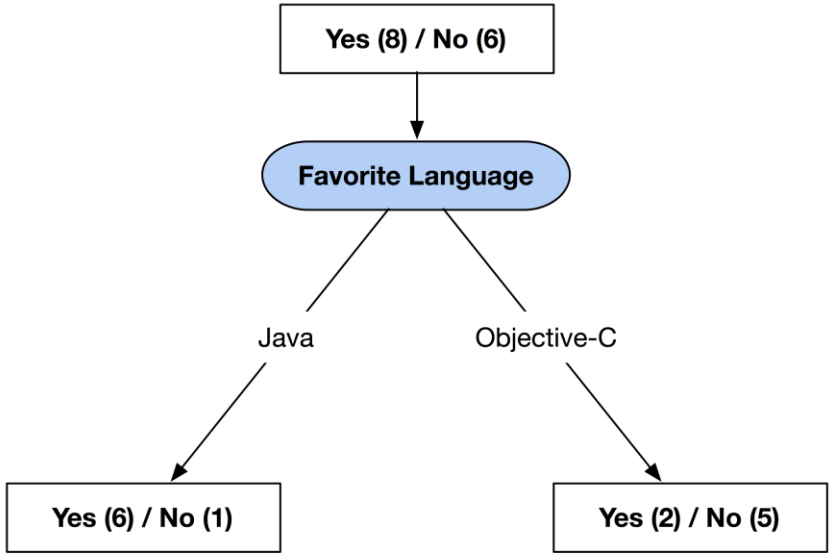
Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985



Entropy(6, 1) = 0.592

Entropy(2, 5) = 0.863

Gain(S, Favorite Language) = 0.985 – (7/14) x 0.592 – (7/14) x 0.863 = 0.258

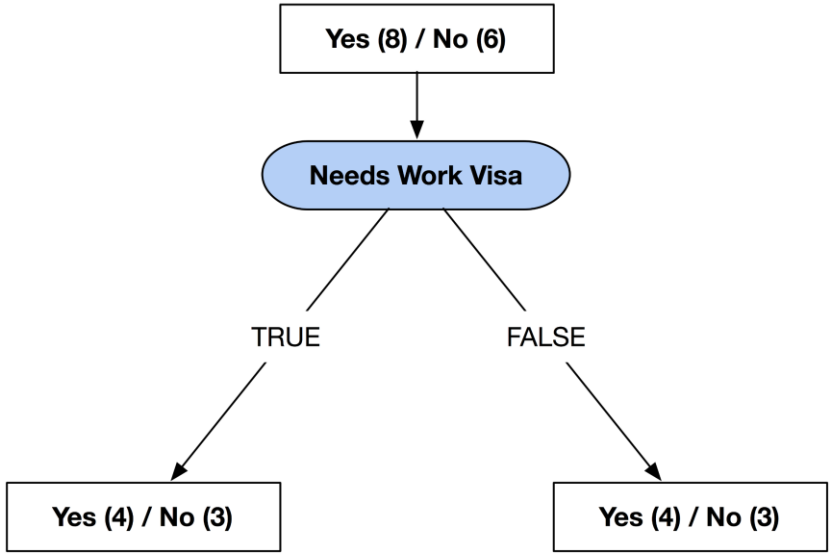
Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Entropy(8, 6) = - (8/14) x log(8/14) - (6/14) x log(6/14) = 0.985

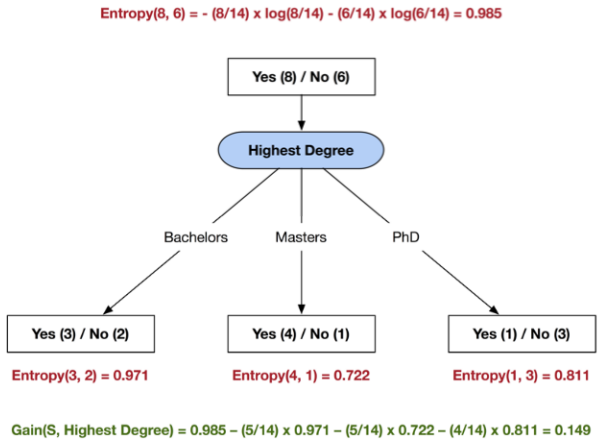


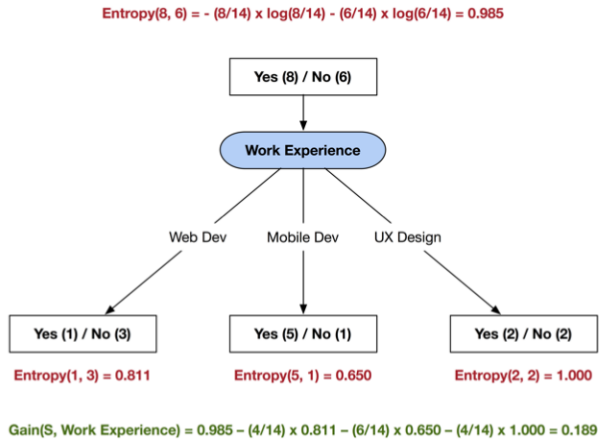
Entropy(4, 3) = 0.985

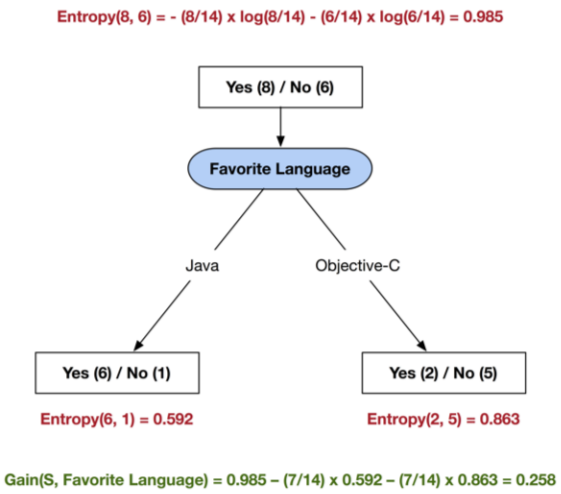
Entropy(4, 3) = 0.985

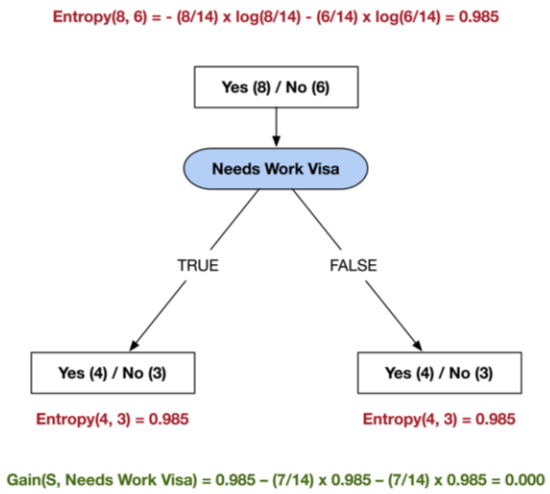
Gain(S, Needs Work Visa) = 0.985 – (7/14) x 0.985 – (7/14) x 0.985 = 0.000

Week 6
Information Theory (Decision Tree – C4.5)









Week 6
Information Theory (Decision Tree – C4.5)

University of Salford MANCHESTER

Feature	Information Gain
Highest Degree	0.149
Work Experience	0.189
Favorite Language	**0.258**
Needs Work Visa	0.000

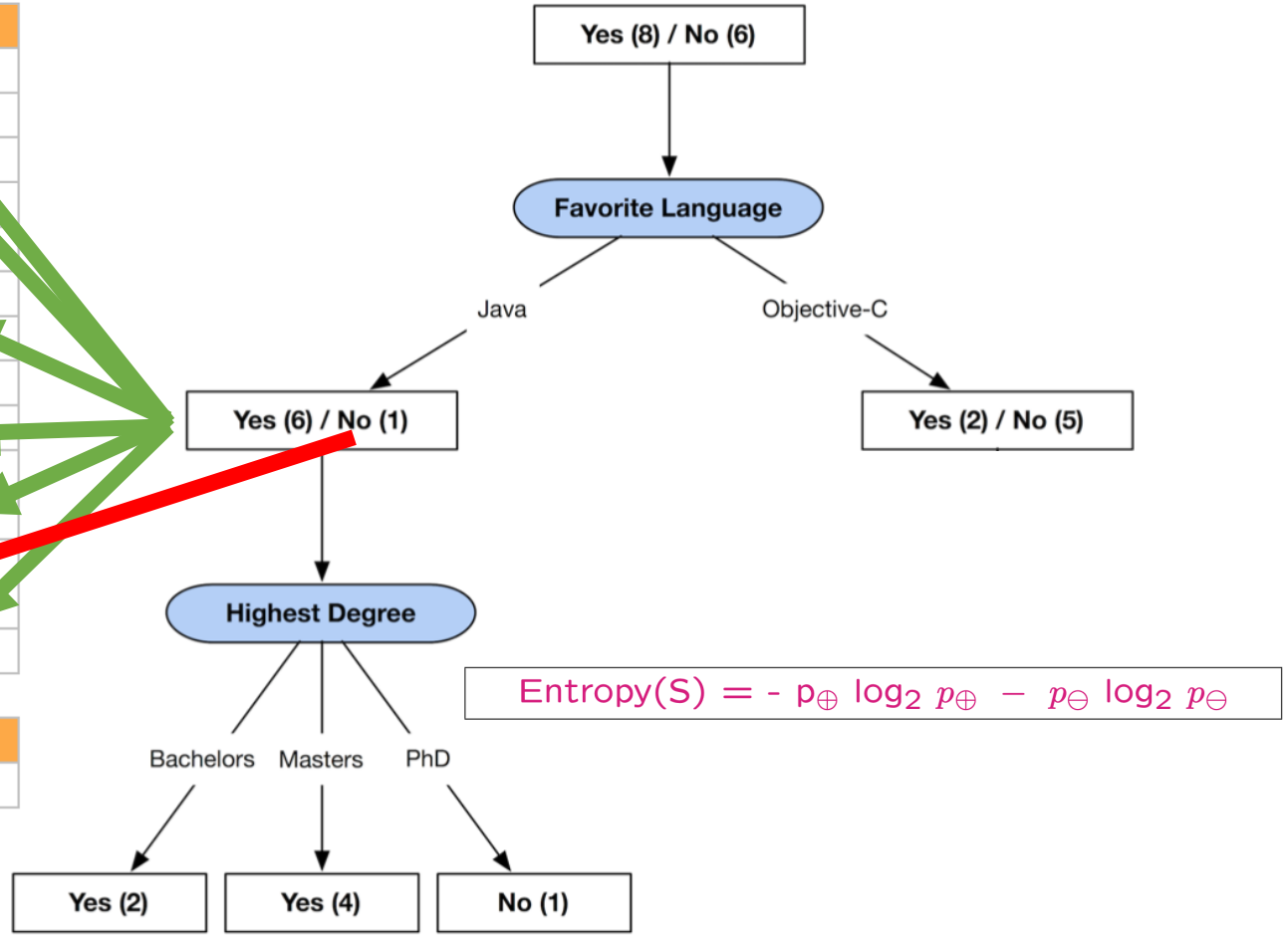
At the first split starting from the root, we choose the attribute that has the max gain.

Then, we re-start the same process at each of the children nodes (if node not pure).

Week 6
Information Theory (Decision Tree – C4.5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?



$$Entropy(S) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in \text{Values(A)}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Week 6
Information Theory (Decision Tree – C4.5)

University of Salford MANCHESTER

```
from sklearn.tree import DecisionTreeClassifier


tree = DecisionTreeClassifier(criterion='entropy')
tree.fit(X_train, y_train)


pridected = tree.predict(X_test)
```

Week 6
Gini Index (Decision Tree – CART/C5)

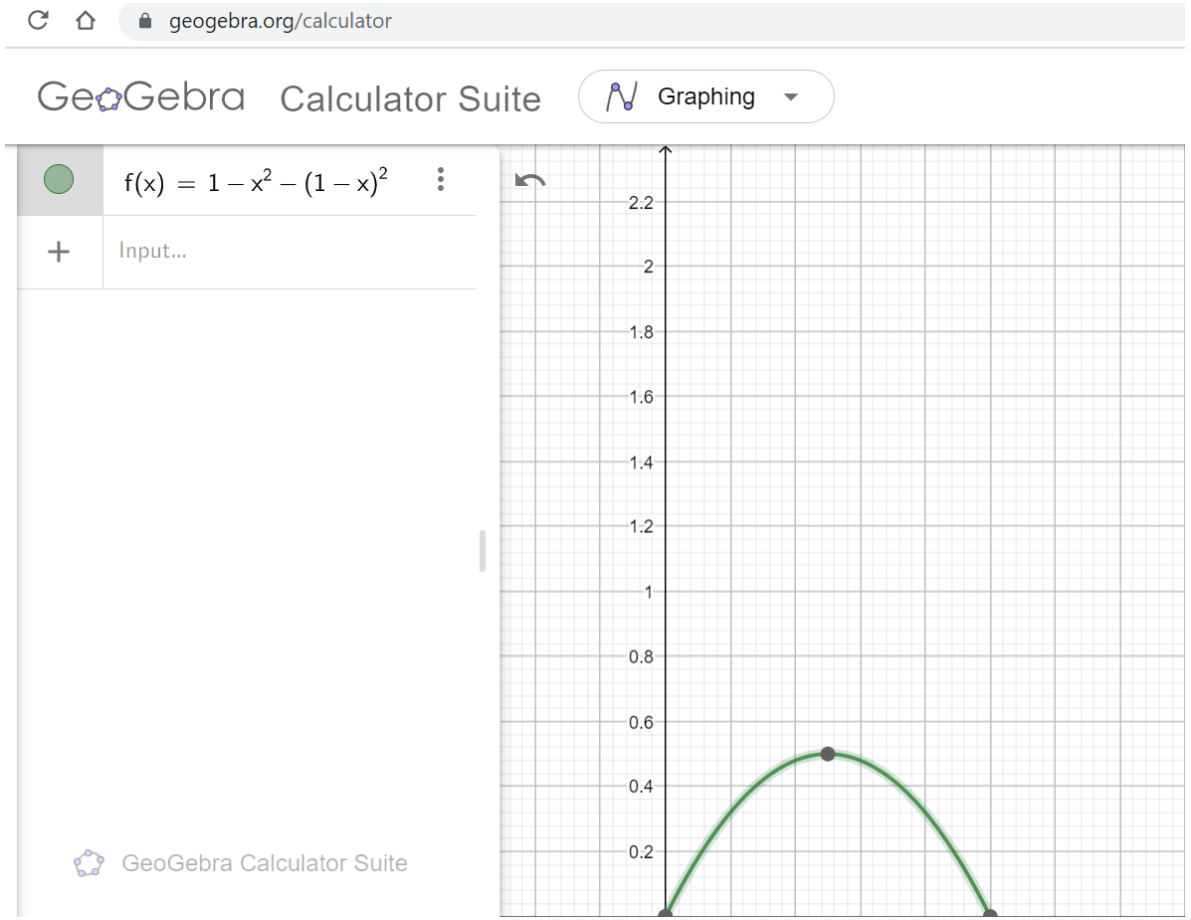
University of Salford MANCHESTER

- Adopt same greedy, top-down algorithm.
- Binary splits instead of multiway splits.
- Uses Gini Index instead of information entropy.

$$Gini = 1 - p_{\oplus}^2 - P_{\ominus}^2$$

Demo

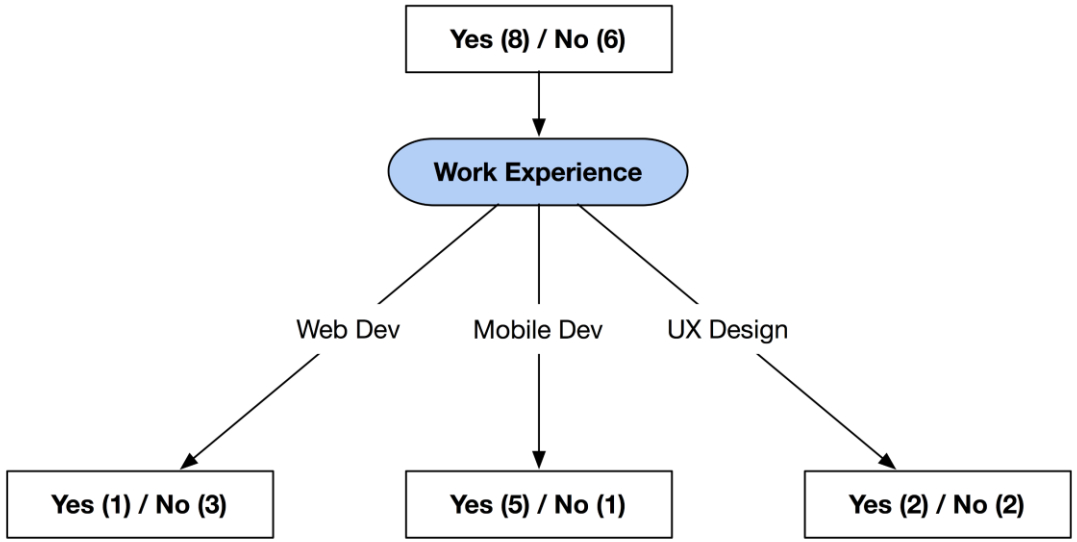
Week 6
Gini Index (Decision Tree – CART/C5)



Week 6
Gini Index (Decision Tree – CART/C5)

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Bachelors	Mobile Dev	Objective-C	TRUE	yes
Masters	Web Dev	Java	FALSE	yes
Masters	Mobile Dev	Java	TRUE	yes
PhD	Mobile Dev	Objective-C	TRUE	yes
PhD	Web Dev	Objective-C	TRUE	no
Bachelors	UX Design	Objective-C	TRUE	no
Bachelors	Mobile Dev	Java	FALSE	yes
PhD	Web Dev	Objective-C	FALSE	no
Bachelors	UX Design	Java	FALSE	yes
Masters	UX Design	Objective-C	TRUE	no
Masters	UX Design	Java	FALSE	yes
PhD	Mobile Dev	Java	FALSE	no
Masters	Mobile Dev	Java	TRUE	yes
Bachelors	Web Dev	Objective-C	FALSE	no

Highest Degree	Work Experience	Favorite Language	Needs Work Visa	Hire
Masters	UX Design	Java	TRUE	?



$$Gini = 1 - p_{\oplus}^2 - P_{\ominus}^2$$

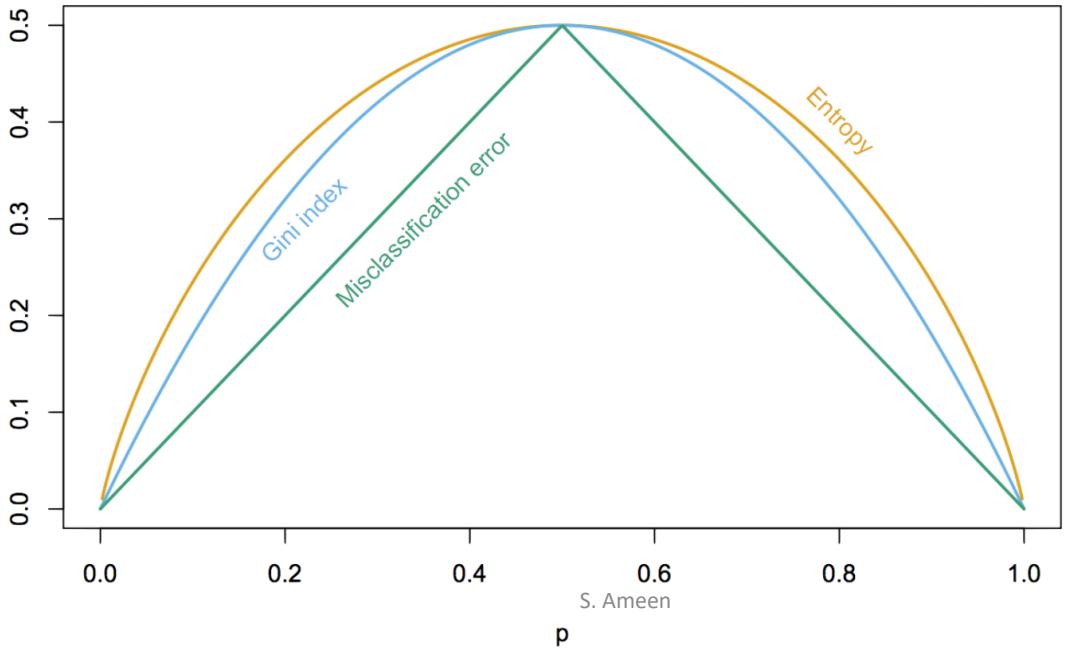
Week 6
Gini Index (Decision Tree – CART/C5)

University of Salford MANCHESTER

```
from sklearn.tree import DecisionTreeClassifier


tree = DecisionTreeClassifier(criterion='gini')
tree.fit(X_train, y_train)


pridected = tree.predict(X_test)
```

Week 6
Comparison of Criteria

- Recall our intuitive guidelines for splitting criteria, which of the three criteria fits our guideline the best?
- We have the following comparison of the value of the three criteria at different levels of purity (from 0 to 1) in a single region (for binary outcomes).



Week 6
Adaptations for Regression

University of Salford MANCHESTER

- With just two modifications, we can use a decision tree model for regression:

1. The three splitting criteria we've examined each promoted splits that were pure - new regions increasingly specialized in a single class.

    A. **For classification**, purity of the regions is a good indicator the performance of the model.
    B. **For regression**, we want to select a splitting criterion that promotes splits that improves the predictive accuracy of the model as measured by, say, the MSE.

2. For regression with output in $\mathbb{R}$, we want to label each region in the model with a real number - typically the average of the output values of the training points contained in the region.

Week 6
Learning Regression Trees

University of Salford MANCHESTER

- The learning algorithms for decision trees in regression tasks is:
  1. Start with an empty decision tree (undivided features pace)
  2. Choose a predictor j on which to split and choose a threshold value t_j for splitting such that the weighted average MSE of the new regions as smallest possible:

$$\underset{j,t_j}{\operatorname{argmin}} \left\{ \frac{N_1}{N} \operatorname{MSE}(R_1) + \frac{N_2}{N} \operatorname{MSE}(R_2) \right\}$$

or equivalently,

$$\underset{j,t_j}{\operatorname{argmin}} \left\{ \frac{N_1}{N} \operatorname{Var}(y|x \in R_1) + \frac{N_2}{N} \operatorname{Var}(y|x \in R_2) \right\}$$

where N_i is the number of training points in R_i and N is the number of points in R.

3. Recurse on each new node until ***stopping condition*** is met.

Week 6
Regression Trees Prediction

University of Salford MANCHESTER

- For any data point x_i

    1. Traverse the tree until we reach a leaf node.
    2. Averaged value of the response variable y's in the leaf (this is from the training set) is the $\widehat{y_i}$.

Week 6
Regression Trees Prediction

University of Salford MANCHESTER

```
from sklearn.tree import DecisionTreeRegressor


tree = DecisionTreeRegressor()
tree.fit(X_train, y_train)


pridected = tree.predict(X_test)
```

Week 6
Variance vs Bias

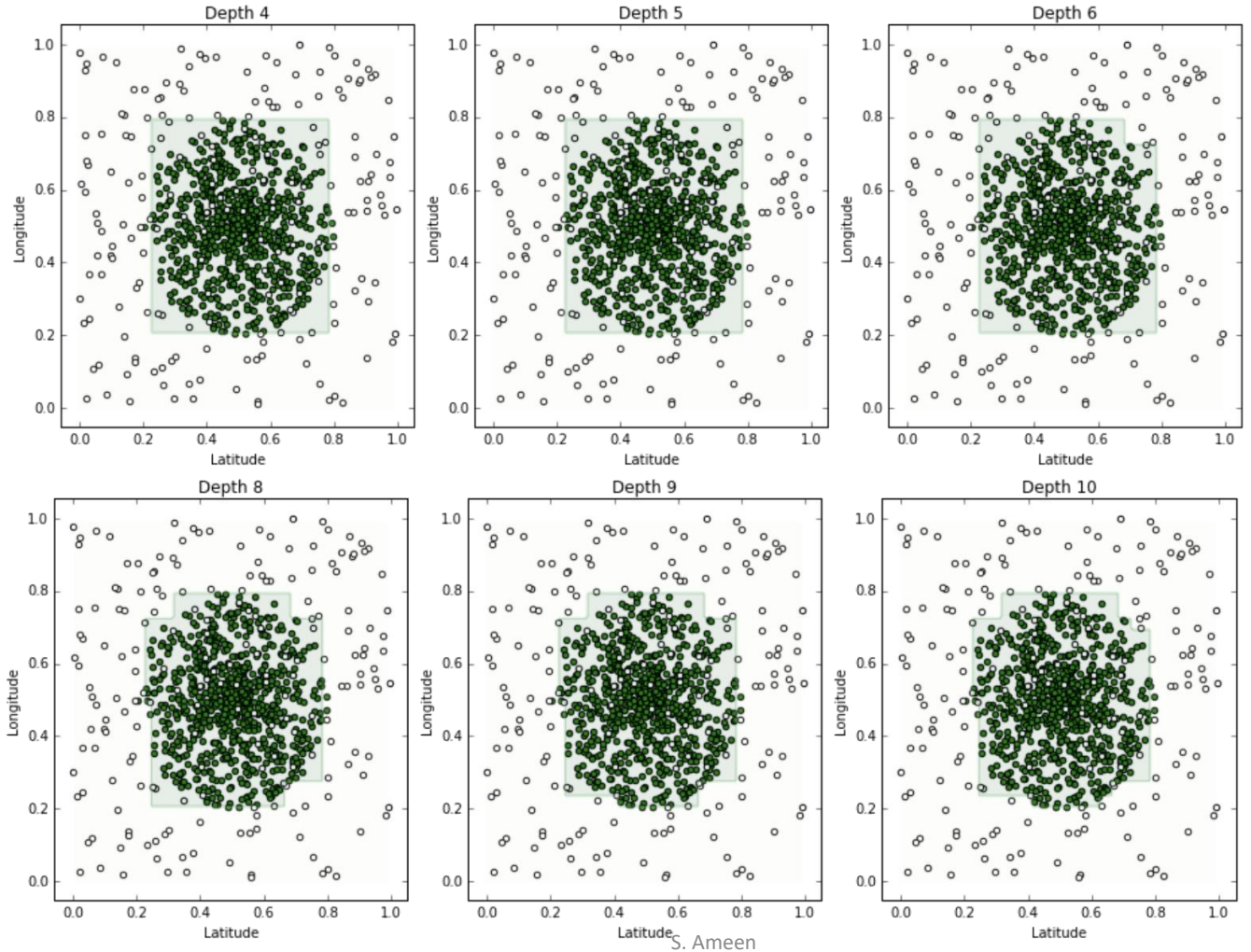
University of Salford MANCHESTER

- If we don’t terminate the decision tree learning algorithm manually, the tree will continue to grow until each region defined by the model possibly contains exactly one training point (and the model attains 100% training accuracy).

- To prevent this from happening, we can simply stop the algorithm at a particular depth.

- But how do we determine the appropriate depth?

Week 6
Variance vs Bias



Week 6
Variance vs Bias

University of Salford MANCHESTER

We make some observations about our models:

- **(High Bias)** A tree of depth 4 is not a good fit for the training data - it's unable to capture the nonlinear boundary separating the two classes.
- **(Low Bias)** With an extremely high depth, we can obtain a model that correctly classifies all points on the boundary (by zig-zagging around each point).
- **(Low Variance)** The tree of depth 4 is robust to slight perturbations in the training data - the square carved out by the model is stable if you move the boundary points a bit.
- **(High Variance)** Trees of high depth are sensitive to perturbations in the training data, especially to changes in the boundary points.
- Not surprisingly, complex trees have low bias (able to capture more complex geometry in the data) but high variance (can overfit). Complex trees are also harder to interpret and more computationally expensive to train.

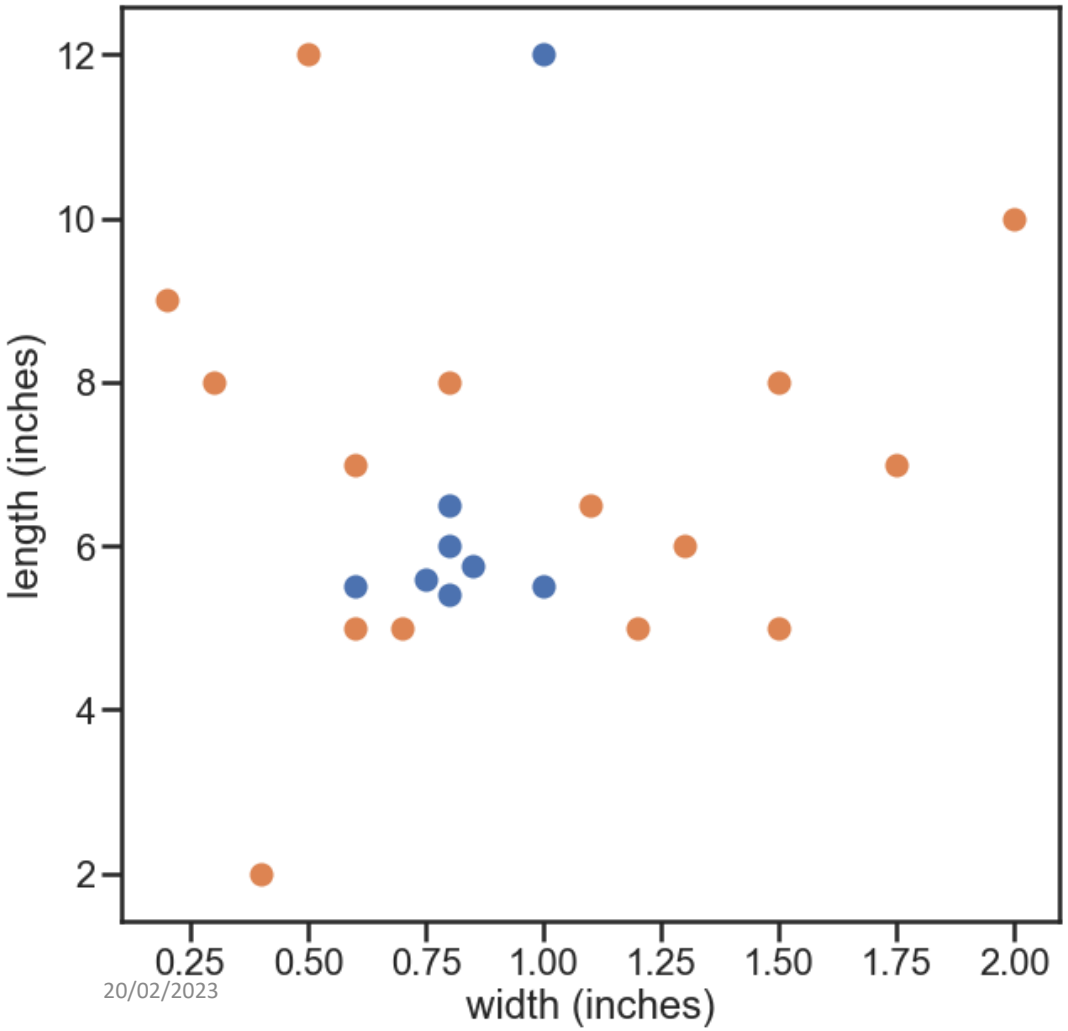
Week 6
Stopping Conditions

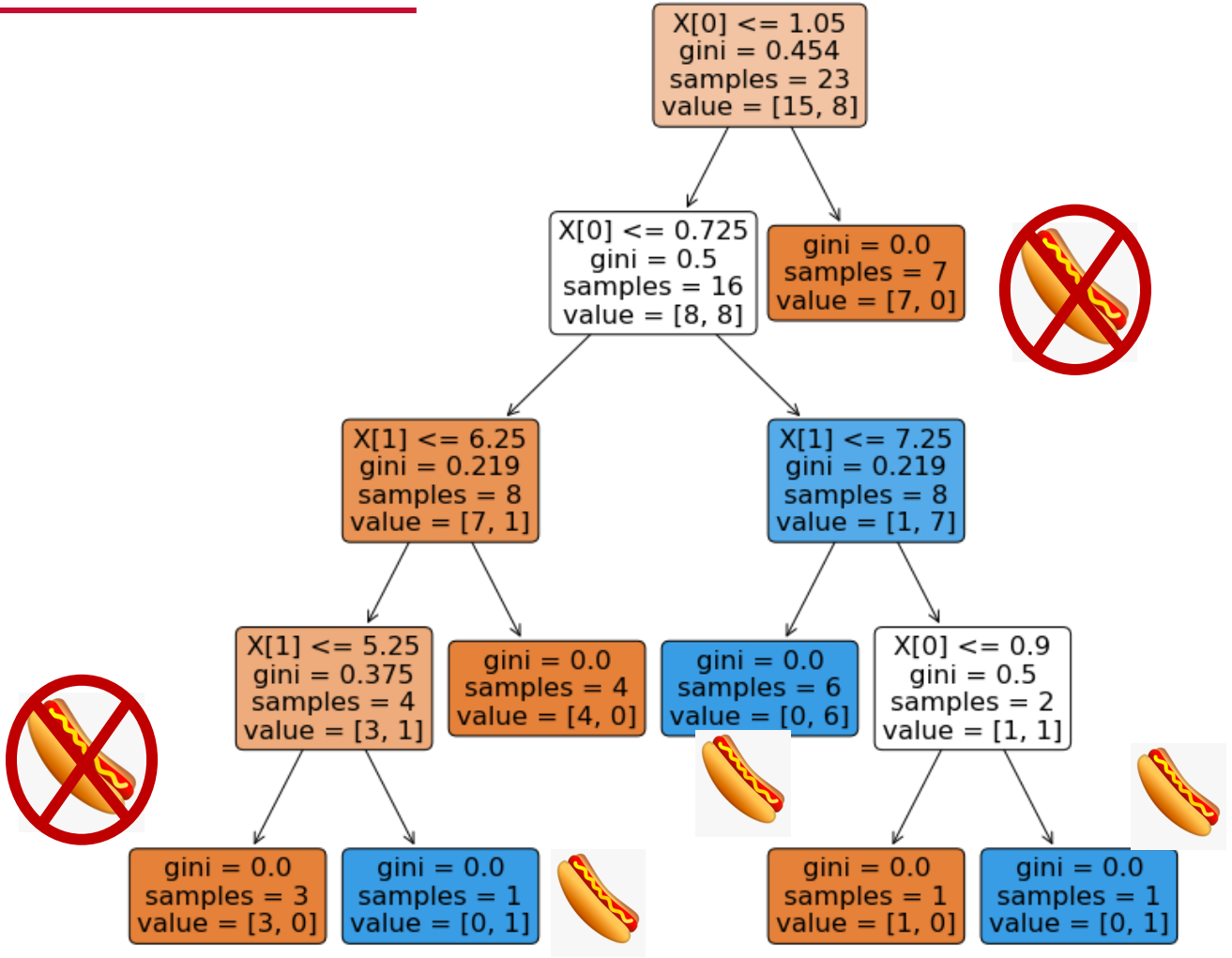
University of Salford MANCHESTER

- Common simple stopping conditions:
  - Don’t split a region if all instances in the region belong to the same class.
  - Don’t split a region if the number of instances in the sub-region will fall below pre-defined threshold (**min_samples_leaf**).
  - Don’t split a region if the total number of leaves in the tree will exceed pre-defined threshold.
- The appropriate thresholds can be determined by evaluating the model on a held-out data set or, better yet, via cross-validation.

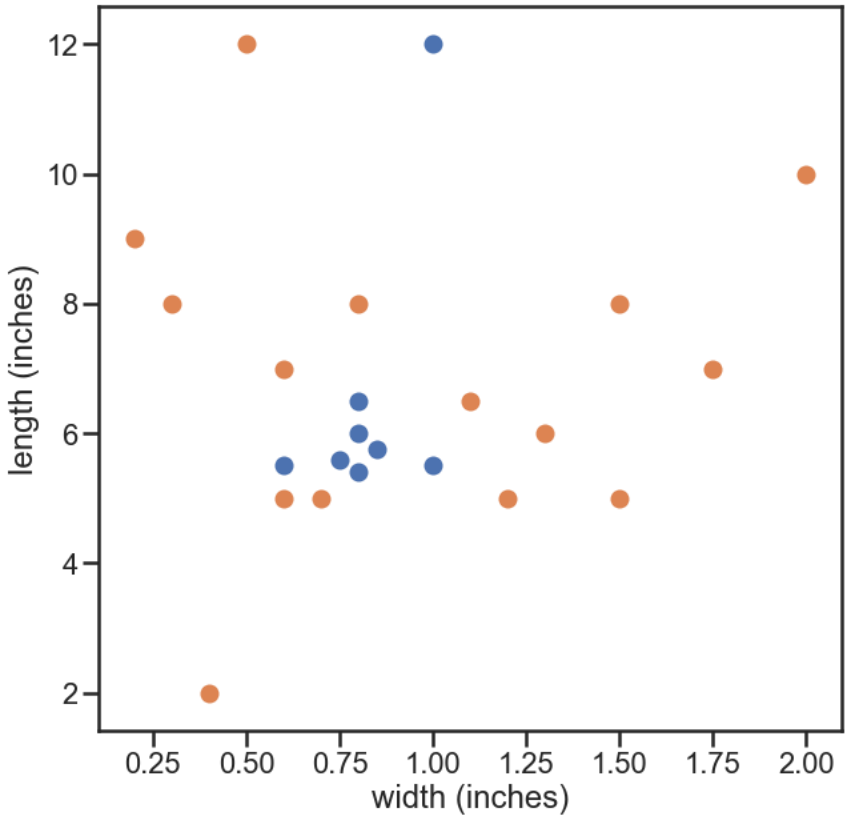
Week 6
To Hot Dog or Not Hot Dog…

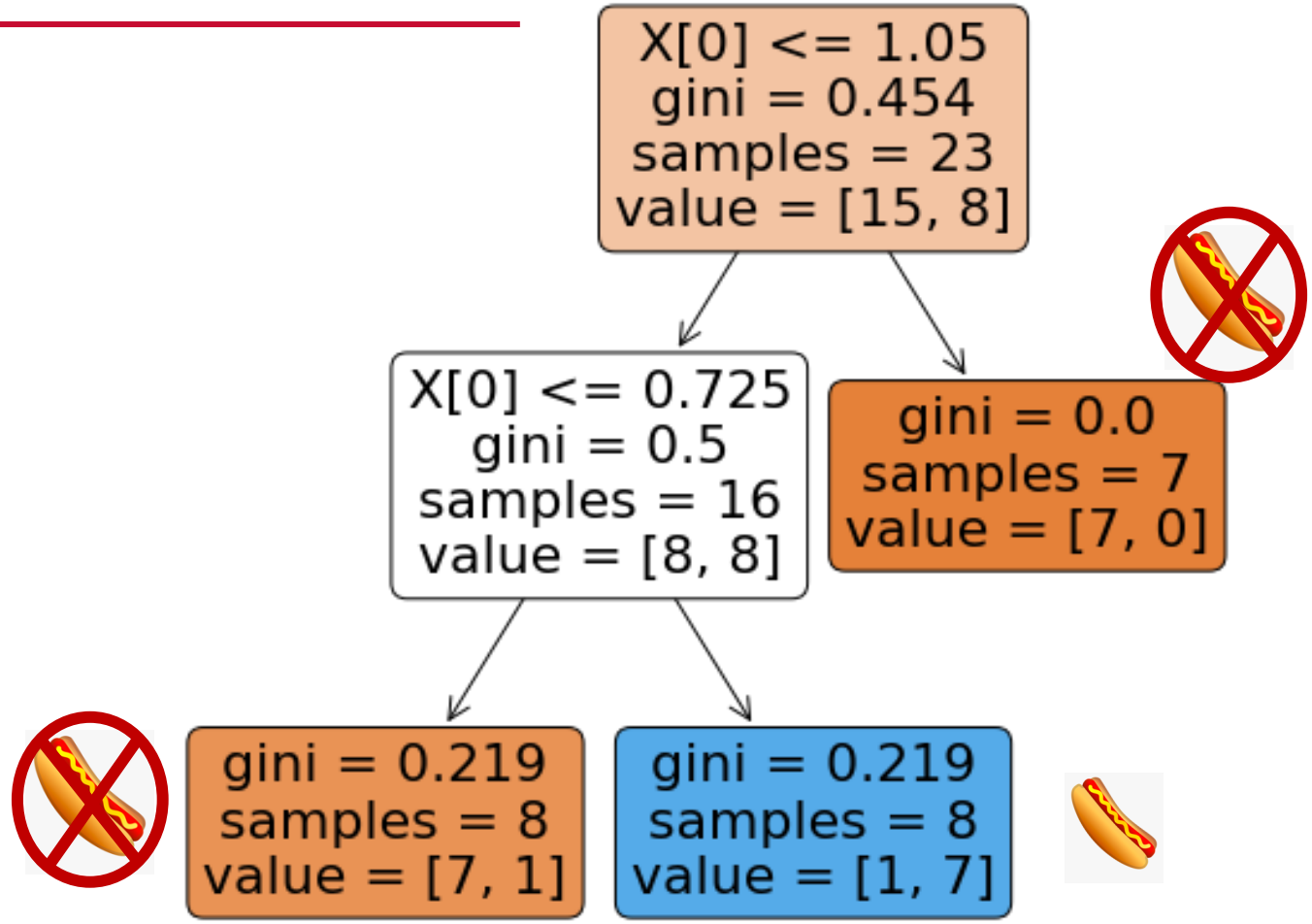
Week 6
Motivation for Pruning



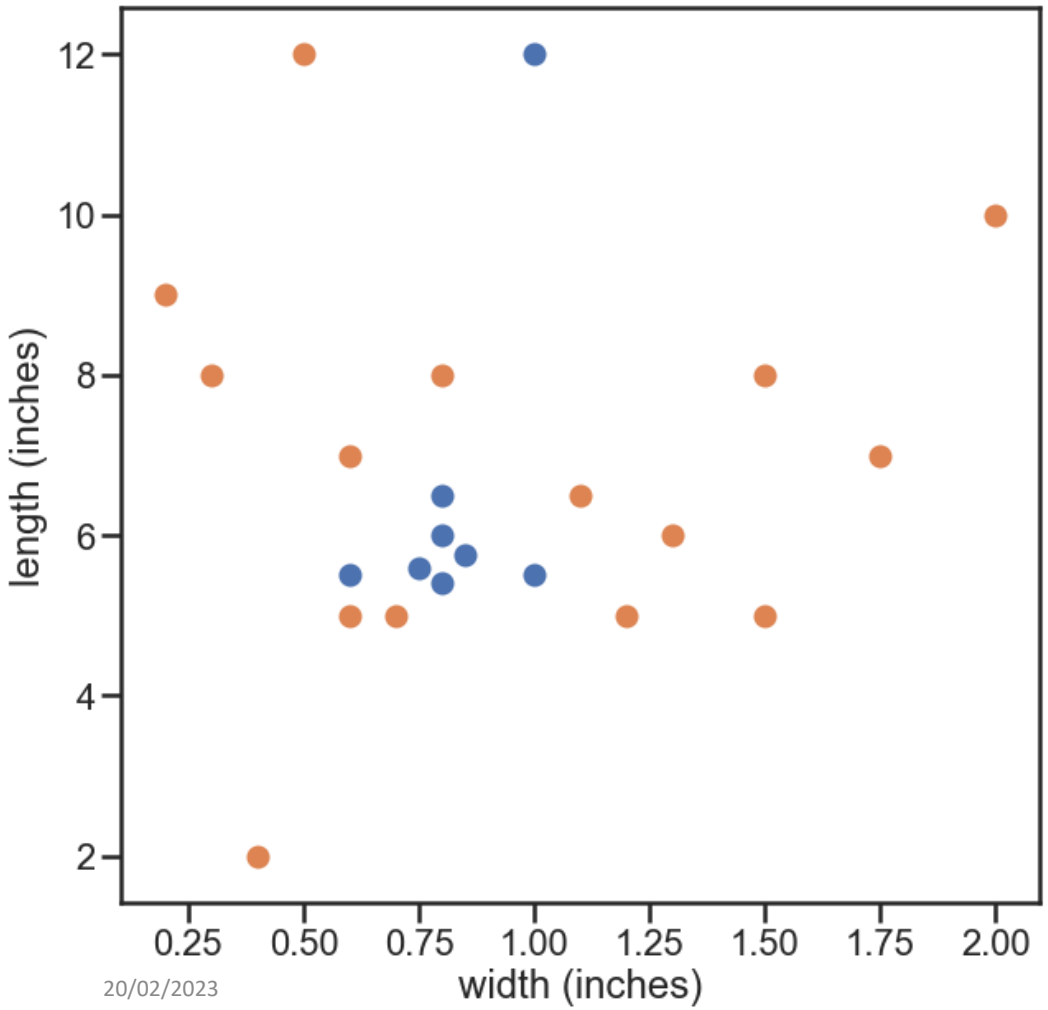


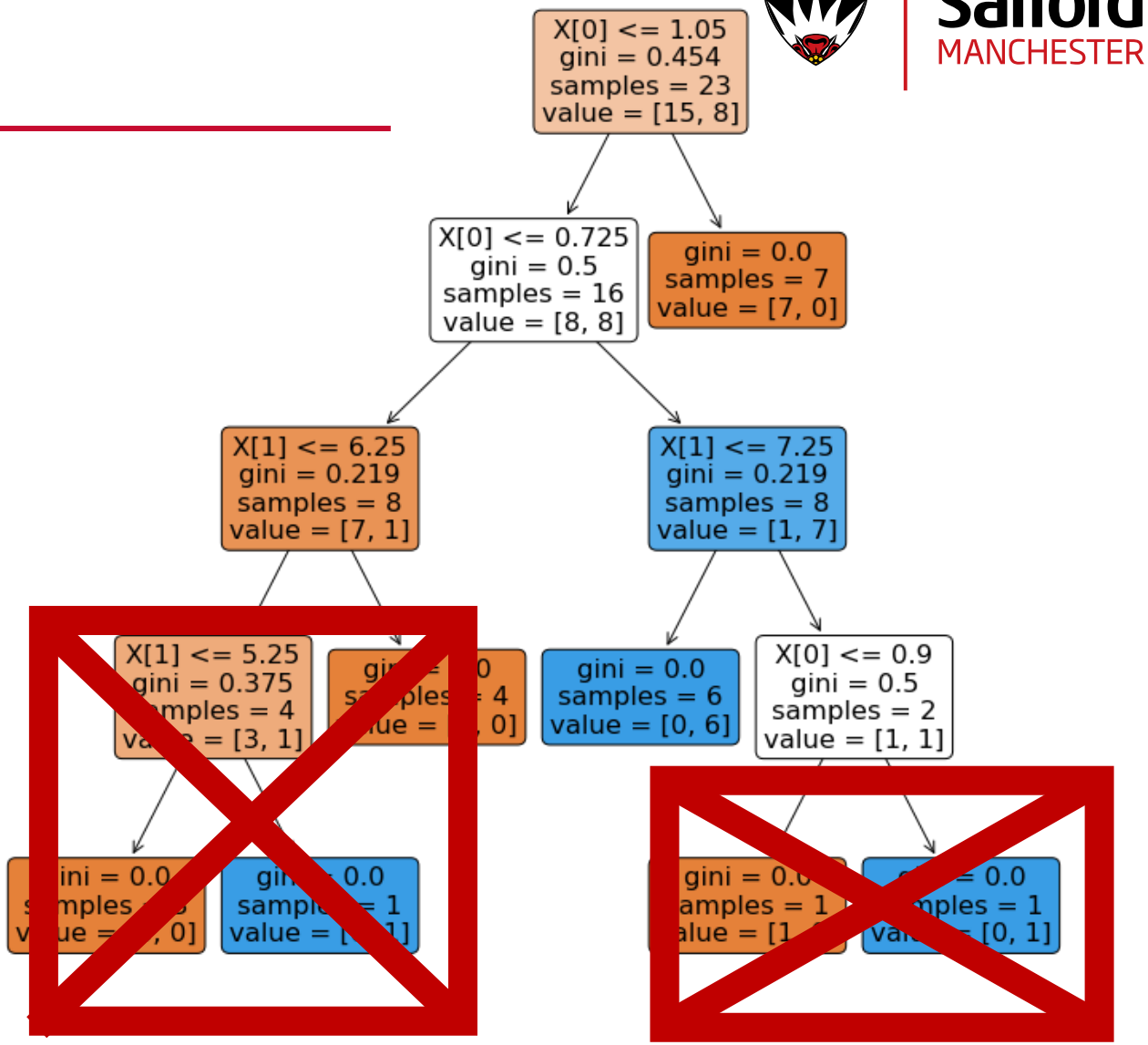
Week 6
Motivation for Pruning



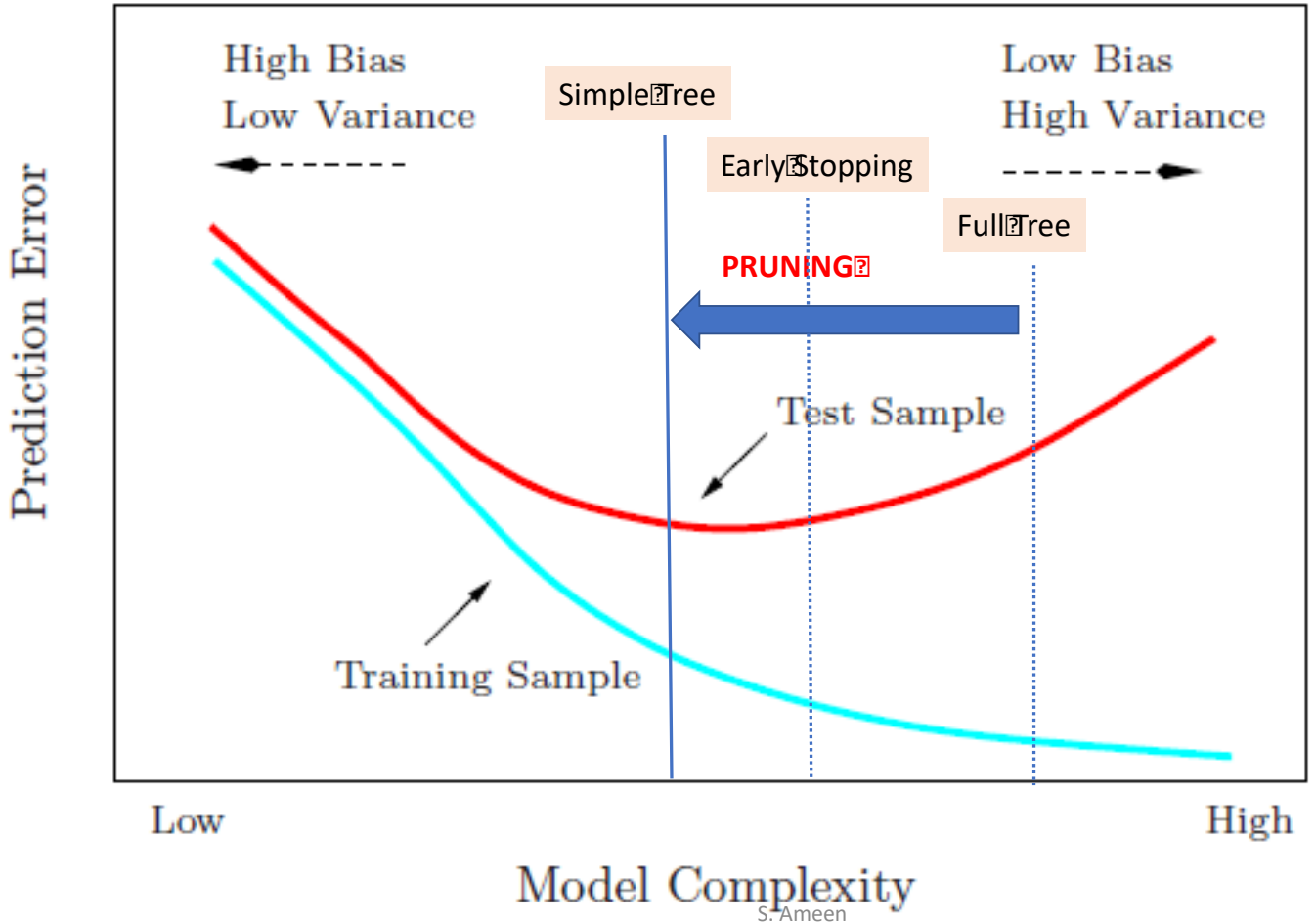


Week 6
Motivation for Pruning





Week 6
Motivation for Pruning



Week 6
Motivation for Pruning

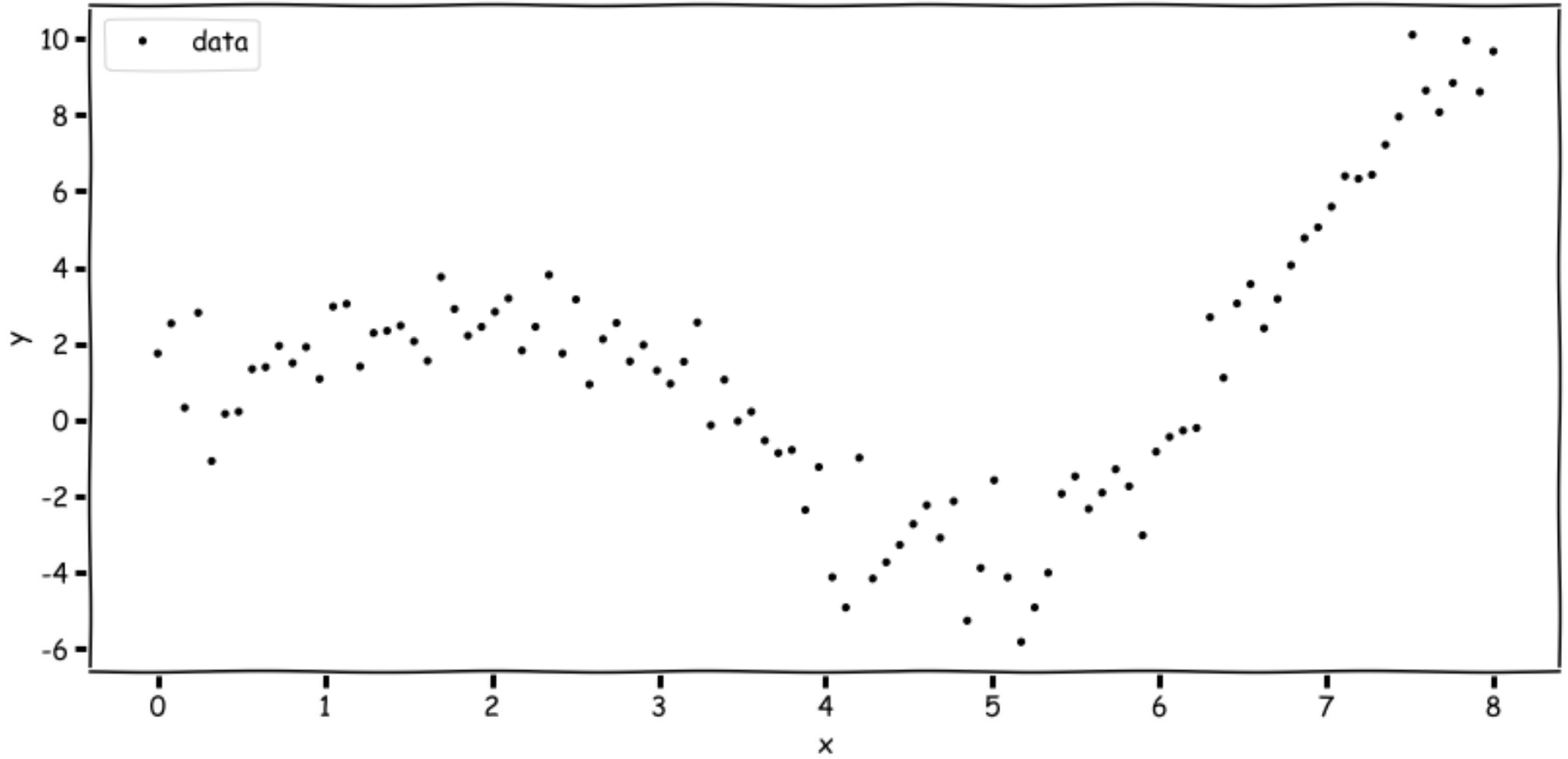

```
# Random Search
parameters =  {'max_depth' : [2,3,4,5,6,7,8,9,10],
               'criterion' : ["gini", "entropy"]}
tree = DecisionTreeClassifier()
clf = RandomizedSearchCV(tree, parameters)

clf.fit(X_train, y_train)
means = clf.cv_results_['mean_test_score']
stds = clf.cv_results_['std_test_score']
for mean, std, params in zip(means, stds, clf.cv_results_['params']):
    print("%0.3f (+/-%0.03f) for %r"
          % (mean, std * 2, params))
```

```
0.504 (+/-0.037) for {'max_depth': 7, 'criterion': 'entropy'}
0.508 (+/-0.026) for {'max_depth': 5, 'criterion': 'entropy'}
0.498 (+/-0.073) for {'max_depth': 9, 'criterion': 'entropy'}
0.525 (+/-0.123) for {'max_depth': 10, 'criterion': 'gini'}
0.490 (+/-0.061) for {'max_depth': 7, 'criterion': 'gini'}
0.519 (+/-0.056) for {'max_depth': 4, 'criterion': 'gini'}
0.502 (+/-0.047) for {'max_depth': 3, 'criterion': 'gini'}
0.508 (+/-0.033) for {'max_depth': 2, 'criterion': 'gini'}
0.508 (+/-0.015) for {'max_depth': 3, 'criterion': 'entropy'}
0.508 (+/-0.046) for {'max_depth': 10, 'criterion': 'entropy'}
```

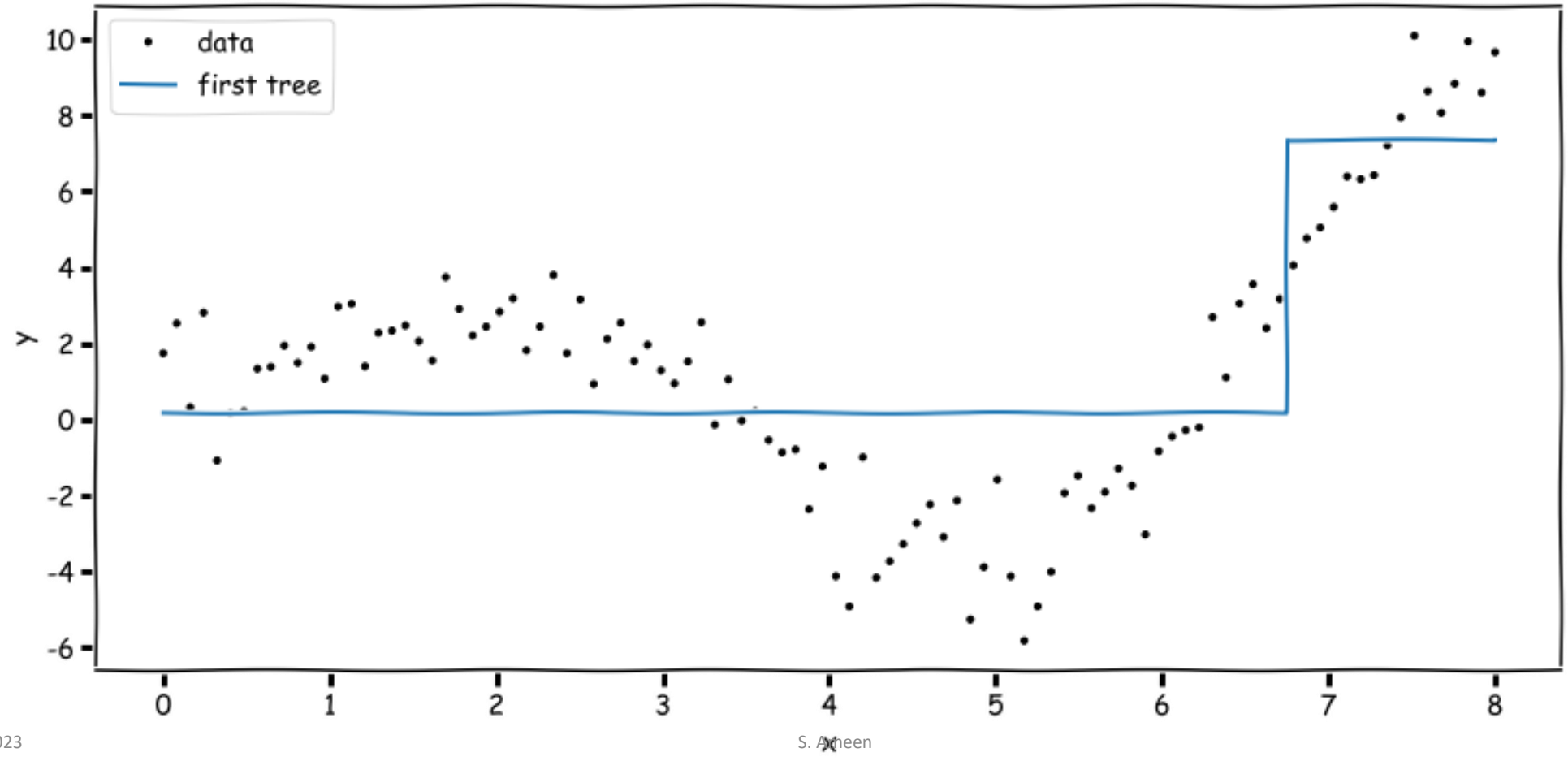
Week 6
Regression Tree Example

University of Salford MANCHESTER

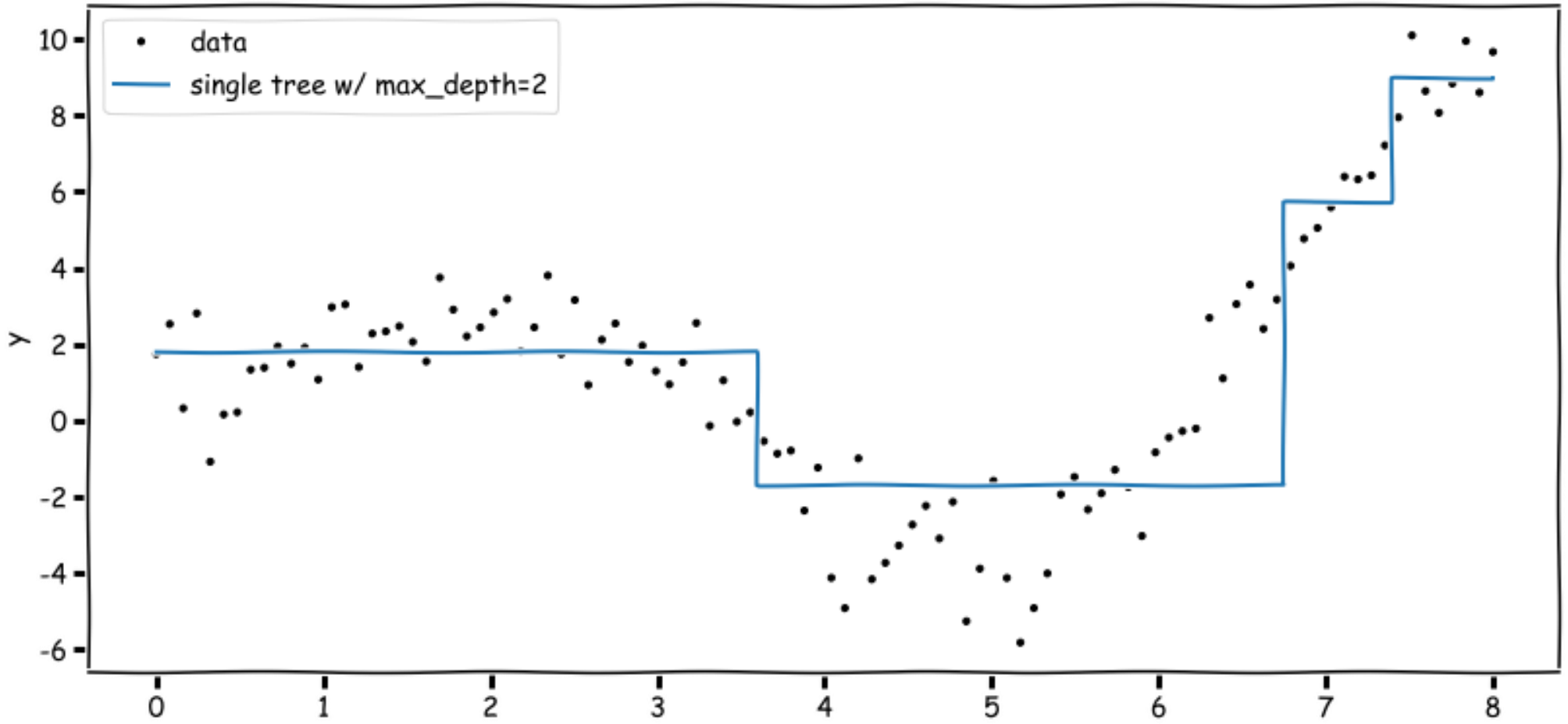
How do we decide a split here?



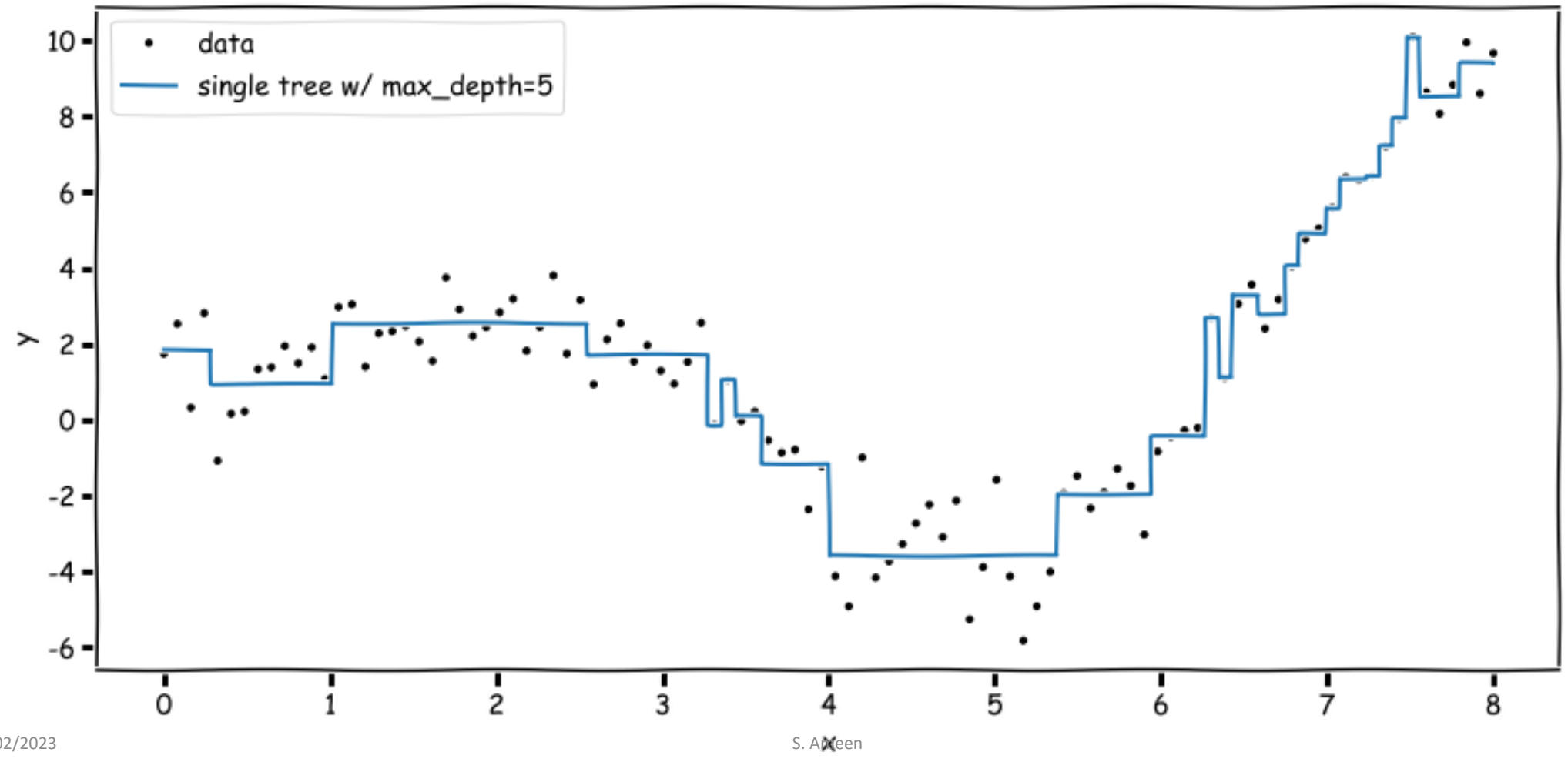
Week 6
Regression Tree (max_depth = 1)



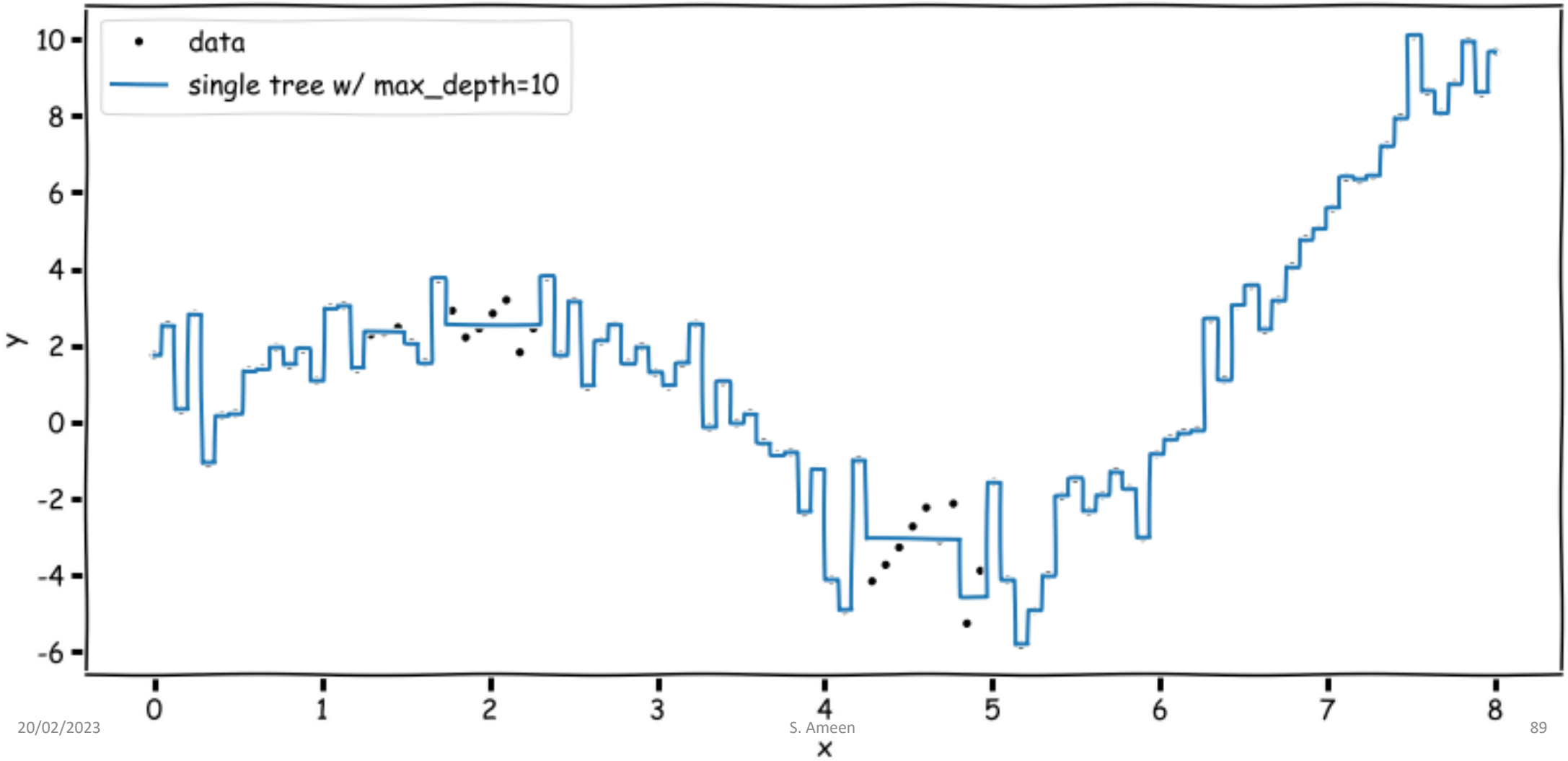
Week 6
Regression Tree (max_depth = 2)



Week 6
Regression Tree (max_depth = 5)

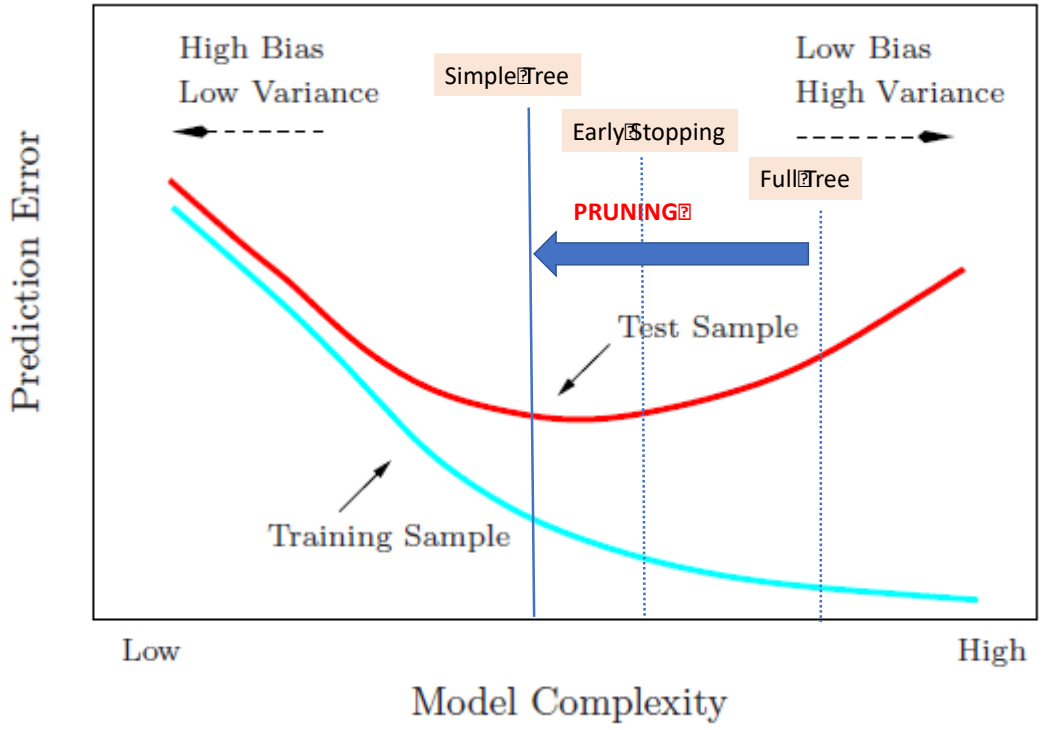


Week 6
Regression Tree (max_depth = 10)



Week 6
Overfitting

- Same issues as with classification trees. Avoid overfitting by pruning or limiting the depth of the tree and using CV.



Week 6
Overfitting

University of Salford MANCHESTER

```
# 2 Random Search

parameters =  {'max_depth' : [2,3,4,5,6,7,8,9,10],
               'criterion' : ["mse", "friedman_mse", "mae"]}
tree = DecisionTreeRegressor()
clf = RandomizedSearchCV(tree, parameters)

clf.fit(X_train, y_train)
means = clf.cv_results_['mean_test_score']
stds = clf.cv_results_['std_test_score']
for mean, std, params in zip(means, stds, clf.cv_results_['params']):
    print("%0.3f (+/-%0.03f) for %r"
          % (mean, std * 2, params))
```

```
0.420 (+/-0.094) for {'max_depth': 10, 'criterion': 'mse'}
0.503 (+/-0.137) for {'max_depth': 7, 'criterion': 'mse'}
0.463 (+/-0.133) for {'max_depth': 8, 'criterion': 'friedman_mse'}
0.597 (+/-0.218) for {'max_depth': 4, 'criterion': 'mse'}
0.455 (+/-0.107) for {'max_depth': 9, 'criterion': 'mae'}
0.597 (+/-0.218) for {'max_depth': 4, 'criterion': 'friedman_mse'}
0.469 (+/-0.147) for {'max_depth': 8, 'criterion': 'mae'}
0.610 (+/-0.192) for {'max_depth': 3, 'criterion': 'mae'}
0.548 (+/-0.118) for {'max_depth': 6, 'criterion': 'mae'}
0.425 (+/-0.111) for {'max_depth': 9, 'criterion': 'friedman_mse'}
```

```
# if you looking for only the best result, instead of last for use the following command
print(clf.best_estimator_)
```

```
DecisionTreeRegressor(criterion='mae', max_depth=3)
```

Week 6 (More on the Workshop)
Numerical vs Categorical Attributes

University of Salford MANCHESTER

Week 6 (More on the Workshop) Hyperparameters

University of Salford MANCHESTER

- Learning rate
- Lambda
- Loss functions
- More

Week 6 : Summary

University of Salford MANCHESTER

- The terminology **Tree** is graphic.
- However, a decision tree is grown from the root downward. The idea is to send the examples down the tree, using the concept of information entropy.
- **General Steps to build a tree:**

  1. Start with the root node that has all the examples.
  2. Greedy selection of the next best feature to build the branches. The splitting criteria is *node purity*.
  3. Class majority will be assigned to the leaves.

Week 6 – Next Lecture
Ensemble Methods - More

University of Salford MANCHESTER

1. Majority voting
2. Boosting
3. Bagging
4. Random Forests

5. SVM
6. Naive Bayes

More