Lecture 8: Supervised learning continued – Linear regression and experimental design

Slides adapted from Prof. James Bailey

Agenda this week

Supervised learning cont.

- Linear regression
- Experimental design
  - Evaluation methods
  - Performance metrics
  - Feature selection

Supervised learning

Techniques for making predictions

- Predict a category or a class (classification)
- Predict a continuous value (regression)

Regression vs. Classification

Classification – Example 1

Predicting a disease from microarray data

Training

	Gene 1	Gene 2	Gene 3	…	Gene n	Develop disease <1 year
Person 1	2.3	1.1	0.3	…	2.1	1
Person 2	3.2	0.2	1.2	…	1.1	1
Person 3	1.9	3.8	2.7	…	0.2	1
…	…	…	…	…	…	..
Person m	2.8	3.1	2.5	…	3.4	0

Testing

	Gene 1	Gene 2	Gene 3	…	Gene n	Develop disease <1 year
Person X	2.1	0.9	0.6	…	1.9	?

Classification – Example 2

Animal classification

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Test data

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Classification – Example 3

Banking: classifying borrowers

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set for predicting borrowers who will default on loan payments.

Test data

Tid	Home Owner	Marital status	Annual Income	Defaulted Borrower
11	No	Single	55K	?

Classification: Definition

Given a collection of records (*training set*)

- Each record contains a set of *attributes*, one *class label*.

Find a predictive *model* for each class label as a function of the values of all attributes, i.e., $y = f(x_1, x_2, \ldots, xn)$

- y: *discrete value*, target variable
- $x_1, \ldots xn$: attributes, predictors
- f: is the predictive model (a tree, a rule, a mathematical formula)

Goal: previously unseen records should be assigned a class as accurately as possible

A *test set* is used to determine the accuracy of the model, i.e. the full data set is divided into training and test sets, with the training set used to build the model and the test set used to validate it

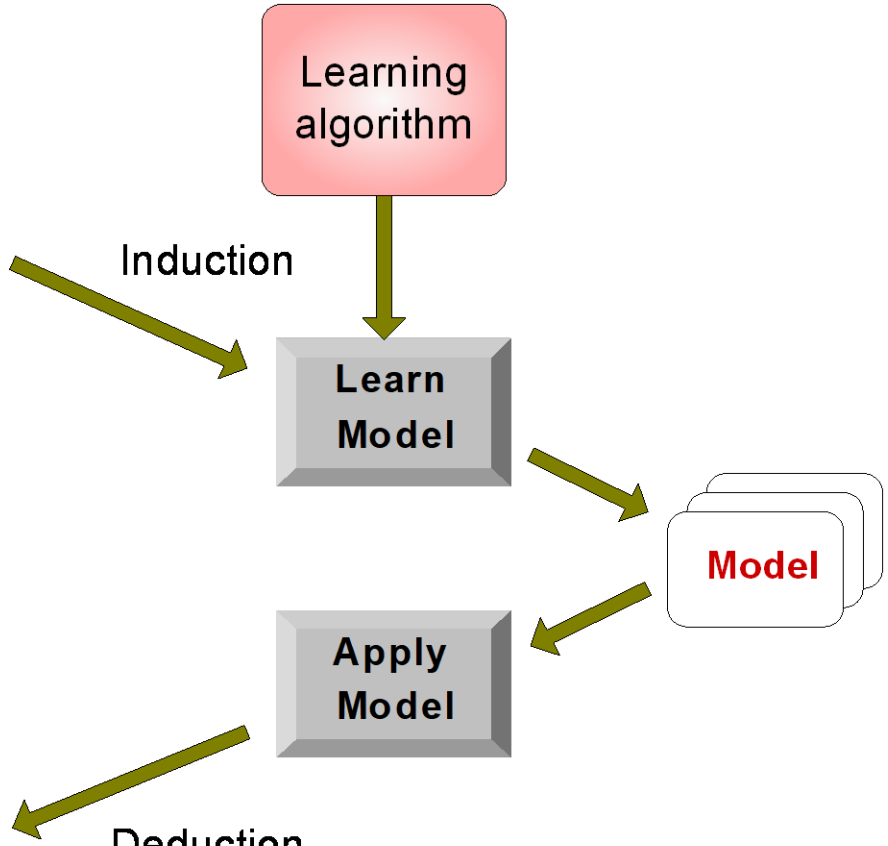
Classification framework

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Regression: Definition

Given a collection of records (*training set*)

- Each record contains a set of *attributes*, one *target variable*.

Find a predictive *model* for each class label as a function of the values of all attributes, i.e., $y = f(x_1, x_2, \ldots, xn)$

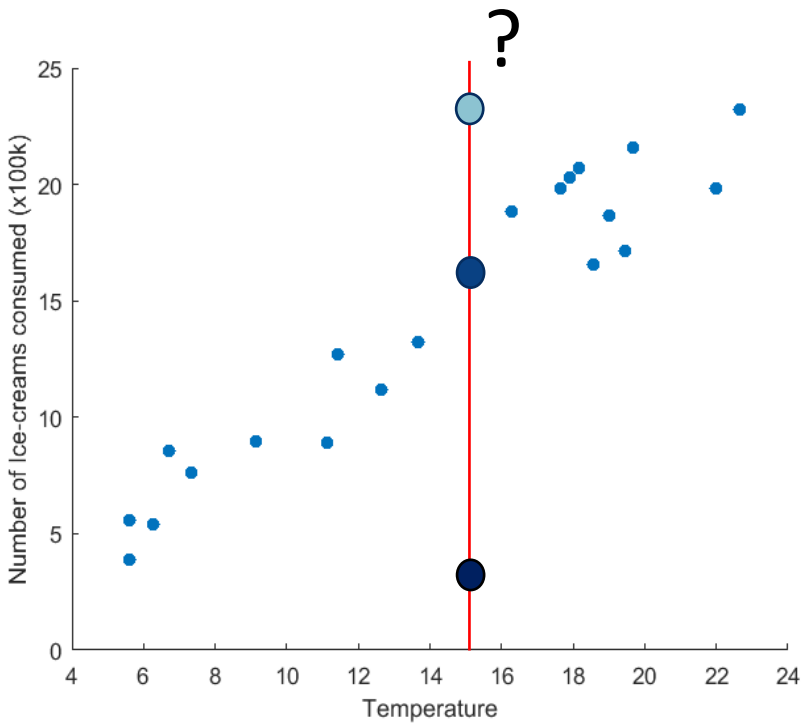
- y: *continuous value*, target variable
- $x_1, \ldots xn$: attributes, predictors
- f: is the predictive model (a tree, a rule, mathematical formula)

Goal: previously unseen records should be assigned a value as accurately as possible

A *test set* is used to determine the accuracy of the model, i.e., the full data set is divided into training and test sets, with the training set used to build the model and the test set used to validate it

Regression example 1

Predicting ice-cream consumption from temperature: $y = f(x)$



Regression example 2

Predicting the activity level of a target gene

	Gene 1	Gene 2	Gene 3	…	Gene n	Gene n+1
Person 1	2.3	1.1	0.3	…	2.1	3.2
Person 2	3.2	0.2	1.2	…	1.1	1.1
Person 3	1.9	3.8	2.7	…	0.2	0.2
…	…	…	…	…	…	…
Person m	2.8	3.1	2.5	…	3.4	0.9

	Gene 1	Gene 2	Gene 3	…	Gene n	Gene n+1
Person m+1	2.1	0.9	0.6	…	1.9	?

Quiz

Is it classification or regression?

- Prediction of rain or no rain tomorrow based on today's rain data
- Prediction of the amount of rain tomorrow based on today's rain data
- Prediction of exam mark based on study hours
- Prediction of exam pass or fail based on study hours
- Prediction of salary based on degree results
- Prediction of complications of surgery based on pre-op medical data

Break

Regression – Linear regression

COMP20008

School of Computing and Information Systems

The following topic uses adapted slides from *Business Statistics: A decision-making approach, Groebner, Shannon and Fry 2018.*

Introduction to Regression Analysis

Regression analysis is used to:

- Predict the value of a dependent variable (Y) based on the value of at least one independent variable (X)
- Explain the impact of changes in an independent variable (X) on the dependent variable (Y)
- Dependent variable (Y): the variable we wish to predict or explain
- Independent variable(s) (X's): the variable(s) used to explain the dependent variable (i.e. the feature(s)/attributes(s))

Notes

(discussions about regression typically use statistics related terminology dependent/independent variable…)

- Quantities which are being predicted or estimated have a ”hat” symbol.
  - E.g. $\widehat{\text{house price}}$ corresponds to a predicted or estimated house price

There is a vast amount of literature on linear regression. In this subject we are approaching from a high-level prediction perspective, not focusing on mathematical rigour.

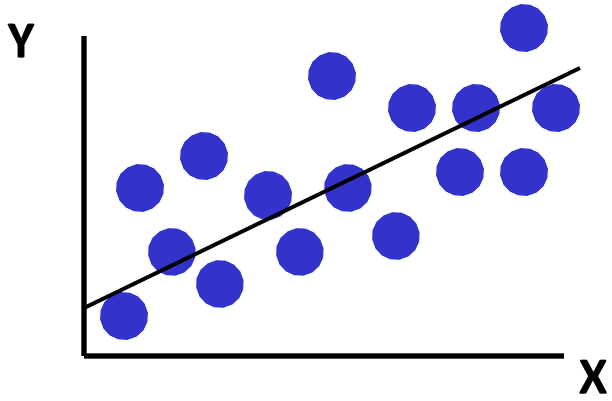
- What are the inputs?
- What is the (predicted) output?
- How do we measure prediction performance?

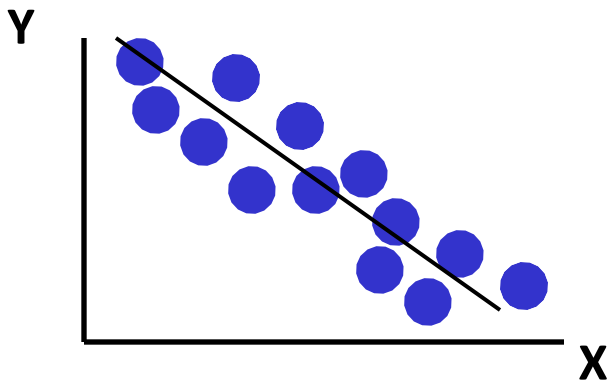
Simple Linear Regression Model

- Only one independent variable (feature), X
- We are trying to predict Y
- Relationship between X and Y is described by a linear function
- Changes in Y are assumed to be caused by changes in X

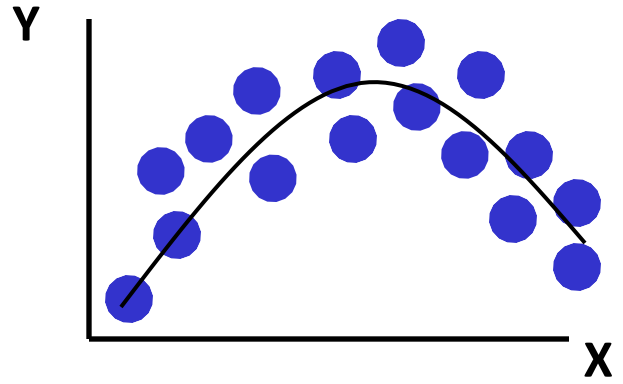
Types of Relationships (1)

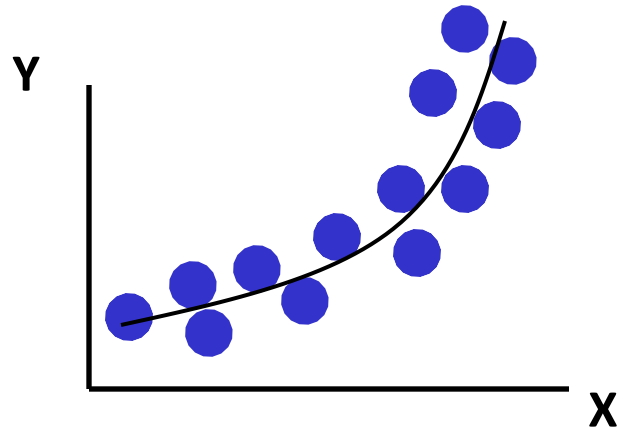
Linear relationships





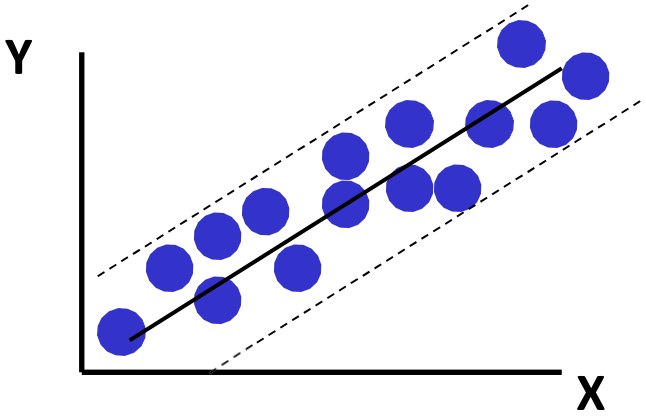
Non-linear relationships

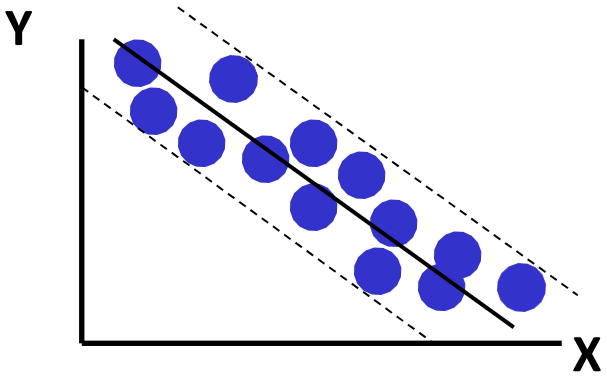




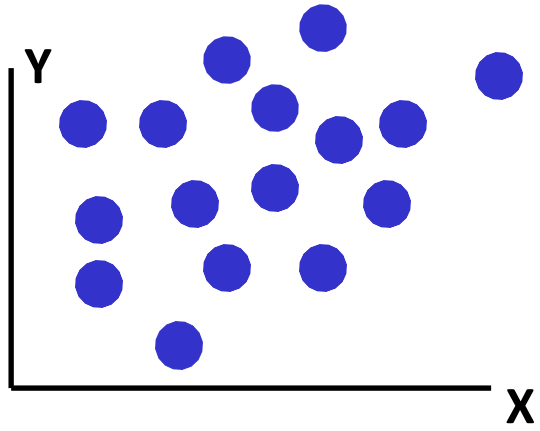
Types of Relationships (2)

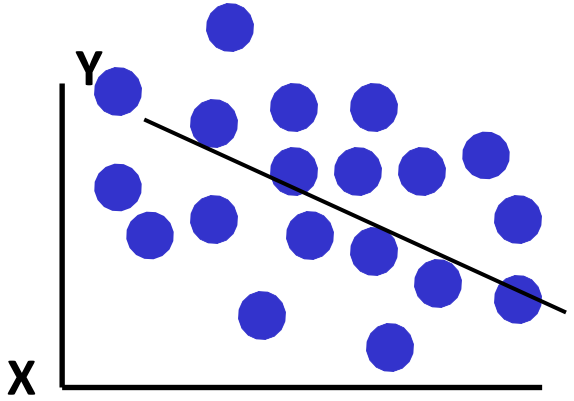
Strong relationships



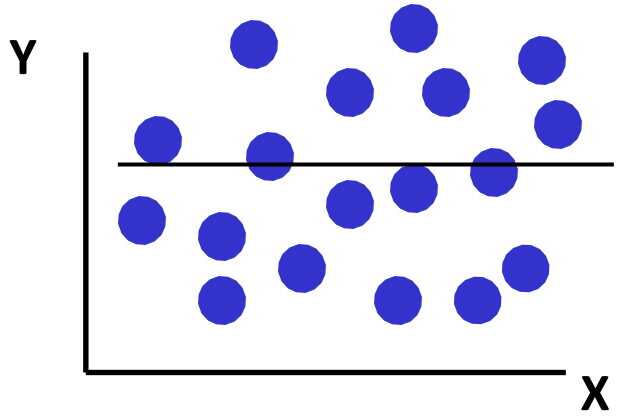


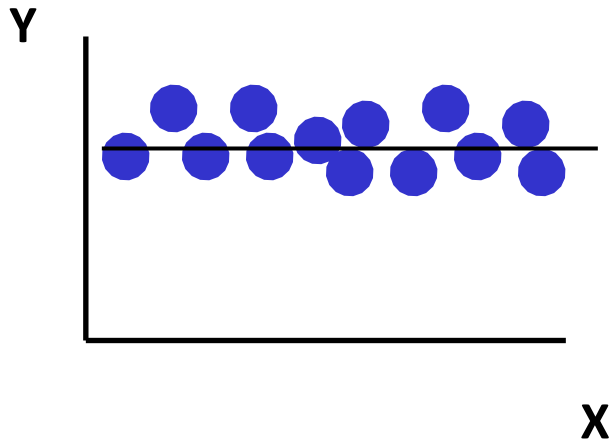
Weak relationships





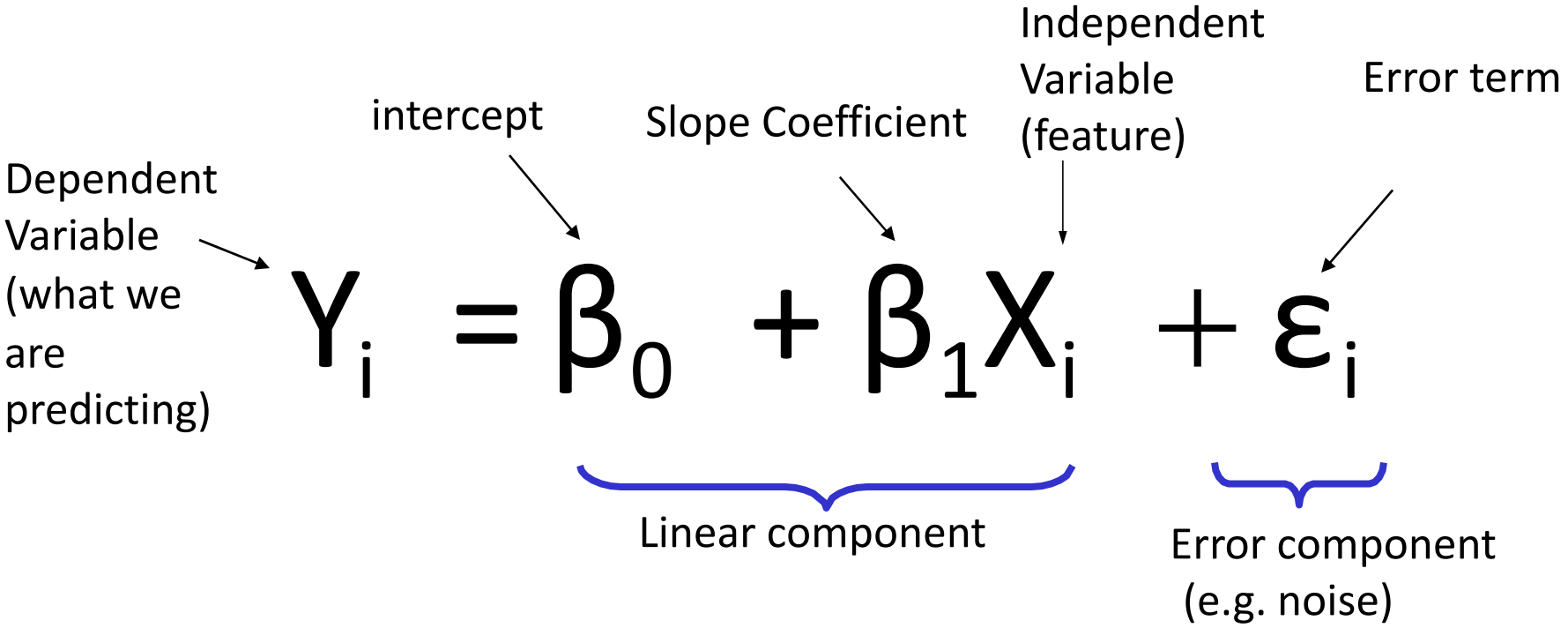
No relationship



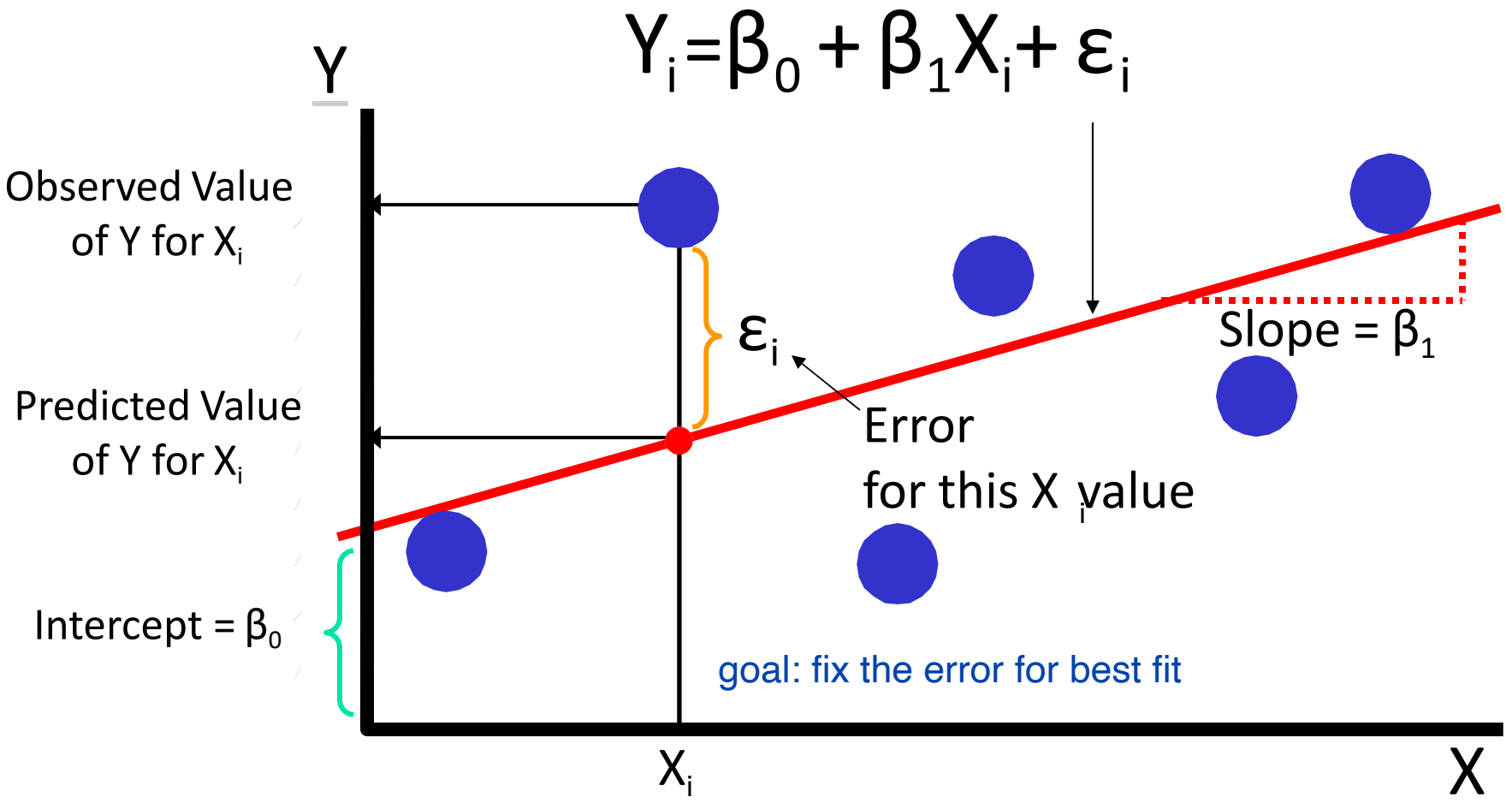


Simple Linear Regression Model

The simple linear regression equation provides an estimate of the population regression line



Simple Linear Regression Model (2)



Least Squares Method

- Used to find the line of best fit

β_0 and β_{1i} are obtained by finding the values of β_0 and β_0 that minimise the sum of the squared differences between Y (true or observed value) and $\hat{Y}$ (predicted value)

$$min \sum \left(Y_i - \hat{Y}_i\right)^2 = min \sum \left(Y_i - (\beta_0 + \beta_1 X_i)\right)^2$$

Interpretation of Slope and Intercept

- β_0 is the estimated average value of Y when the value of X is zero (intercept)
  - More intuitively how high or low the regression line sits on the graph, where vertical starting point is X = 0

- β_1 is the estimated change in the average value of Y as a result of a one-unit change in X (slope)
  - More intuitively how tilted is the line (is it flat, steep?)

Simple Linear Regression Example

A real estate agent wishes to examine the relationship between the selling price of a home and its size (USA example --- measured in square feet)

A random sample of 10 houses is selected

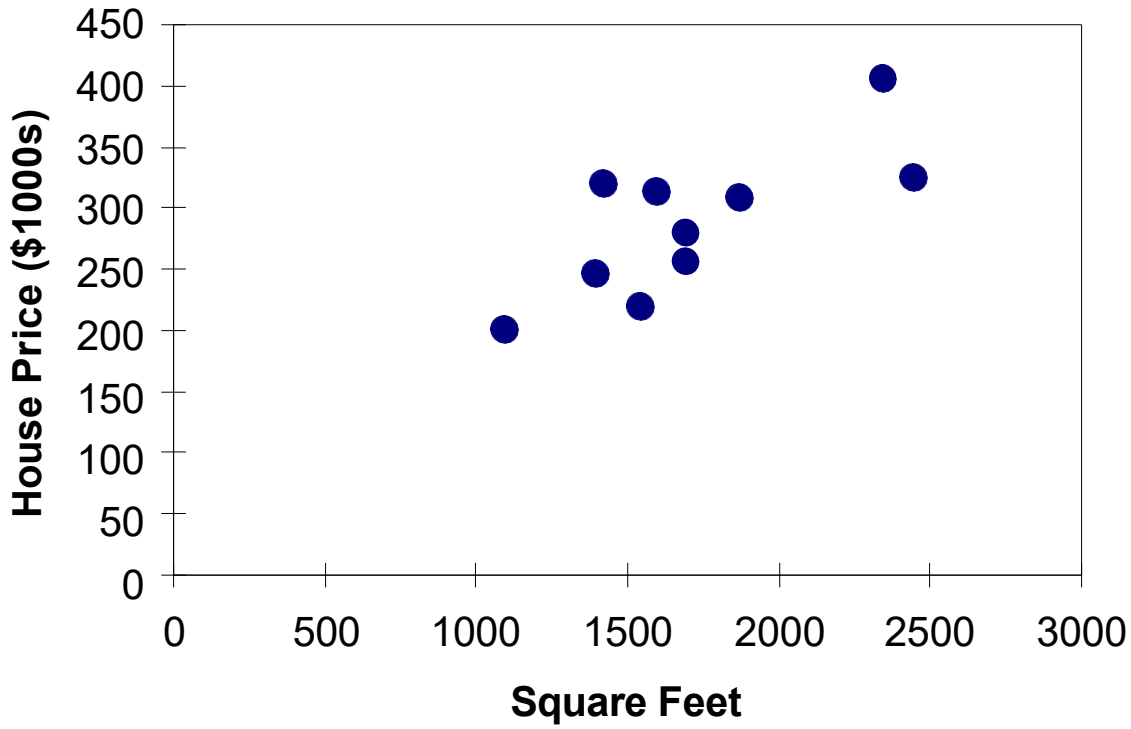
- Dependent variable (Y) = house price in $1000s
- Independent variable (X) = square feet

Sample Data for House Price Model

House Price in $1000s (Y)	SquareFeet (X)
245	1400
312	1600
279	1700
308	1875
199	1100
219	1550
405	2350
324	2450
319	1425
255	1700

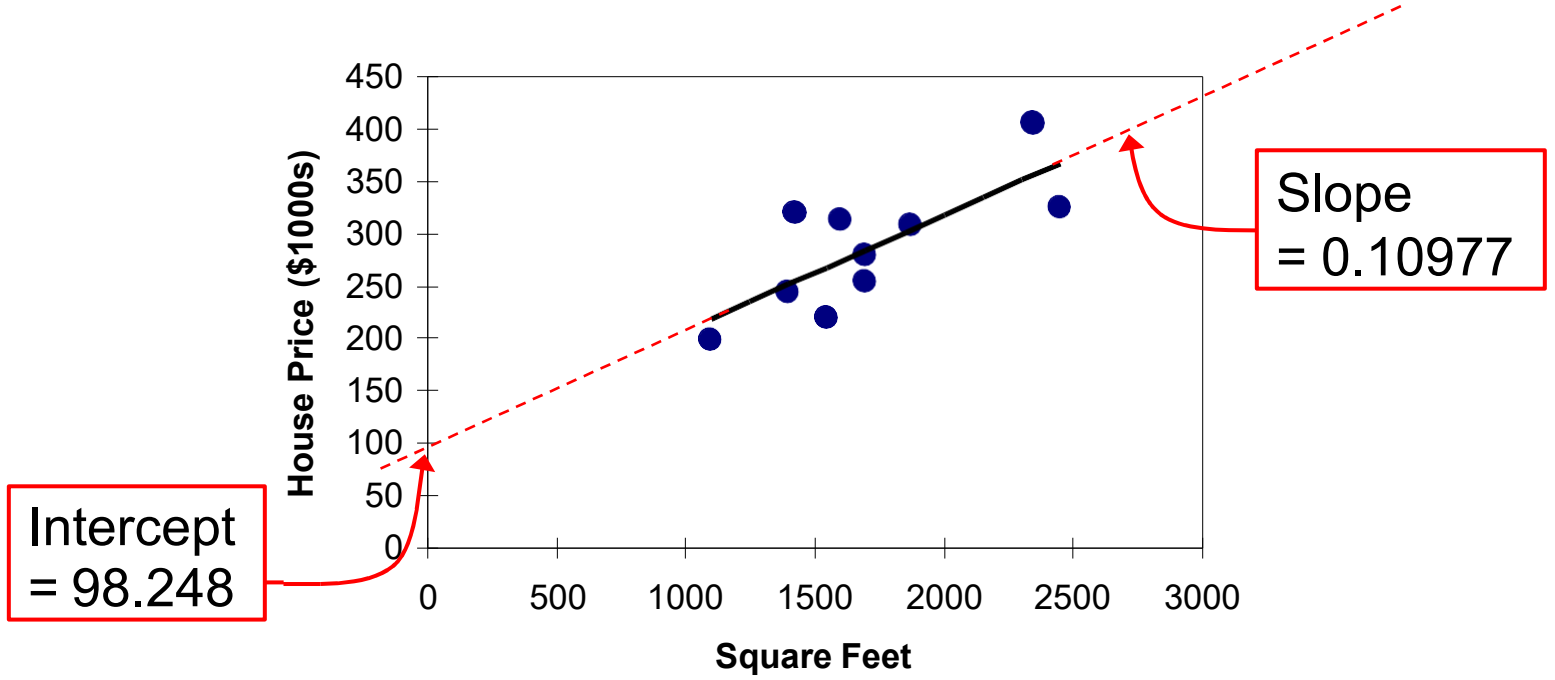
Graphical Representation

House price model: scatter plot



Graphical Representation

House price model: scatter plot and regression line



$$\widehat{\text{house price}} = 98.24833 + 0.10977\,(\text{square feet})$$

Interpretation of the Intercept β_0

$$\widehat{\text{house price}} = 98.24833 + 0.10977\ (\text{square feet})$$

- β_0 is the estimated average value of Y when the value of X is zero
- Here, no houses had 0 square feet, so β_0 = 98.24833 ($1000) just indicates that, for houses within the range of sizes observed, $98,248.33 is the portion of the house price not explained by square feet

Interpretation of the Slope Coefficient β_1

$$\widehat{\text{house price}} = 98.24833 + 0.10977\,(\text{square feet})$$

- β_1 measures the estimated change in the average value of Y as a result of a one-unit change in X
- Here, β_1 = .10977 tells us that the average value of a house increases by .10977 ($1000) = $109.77, on average, for each additional one square foot of size

Predictions using Regression

Predict the price for a house with 2000 square feet:

$$\widehat{\text{house price}} = 98.25 + 0.1098\ (\text{sq.ft.})$$

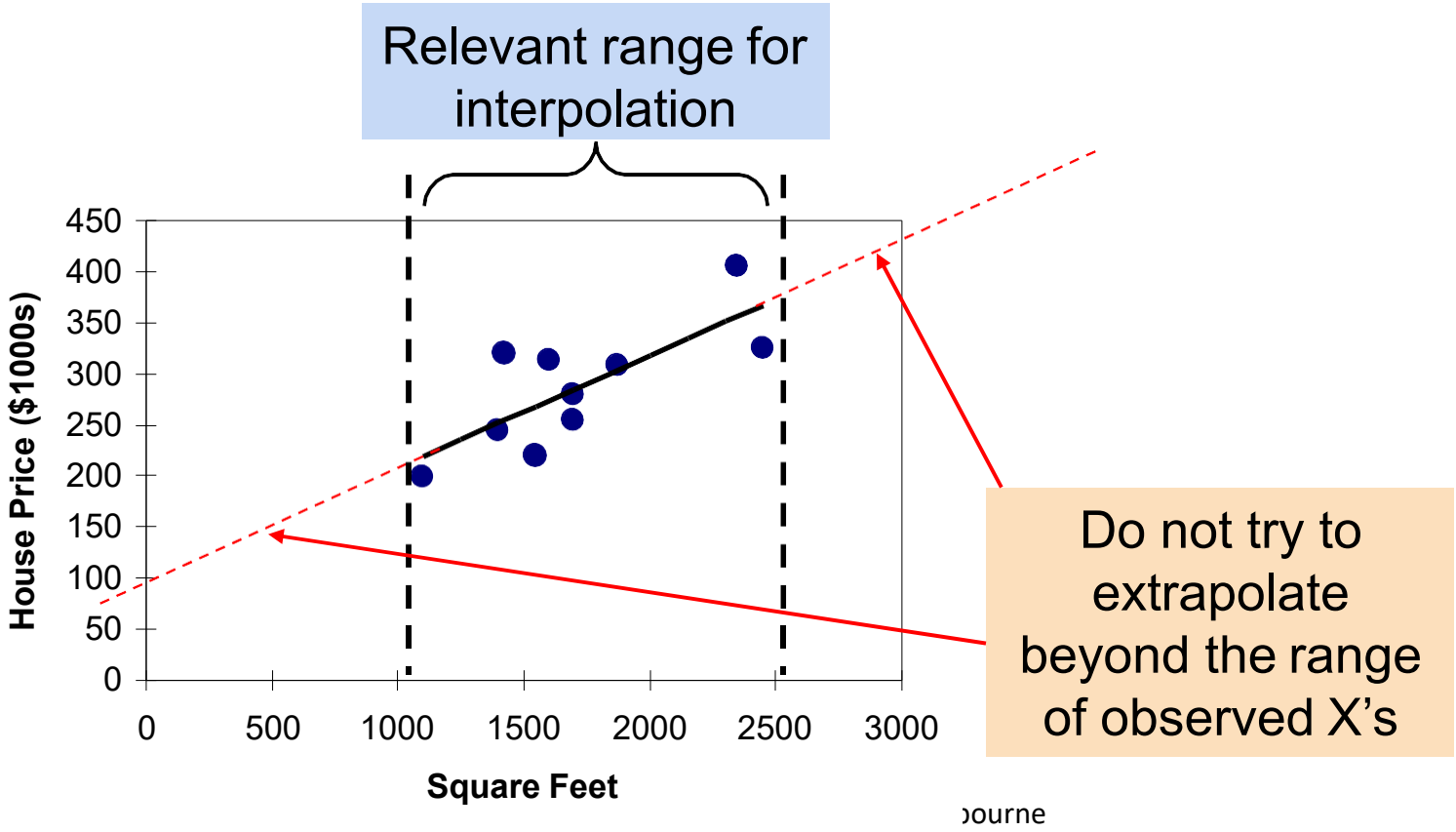
$$= 98.25 + 0.1098\ (2000)$$

$$= 317.85$$

The predicted price for a house with 2000 square feet is 317.85 ($1,000s) = $317,850

Interpolation vs. Extrapolation

Guideline: When using a regression model for prediction, only predict within the relevant range of data



Multiple Regression

- Multiple regression is an extension of simple linear regression
- It is used when we want to predict the value of a variable based on the value of two or more other variables
- The variable we want to predict is called the dependent variable
- The variables we are using to predict the value of the dependent variable are called the independent variables

Multiple Regression Example

A researcher may be interested in the relationship between the weight of a car, the power of the engine, and petrol consumption.

Independent Variable 1: weight

Independent Variable 2: horsepower

Dependent Variable: miles per gallon

Multiple Regression Fitting

- Linear regression is based on fitting a line as close as possible to the plotted coordinates of the data on a two-dimensional graph
- Multiple regression with two independent variables is based on fitting a plane as close as possible to the plotted coordinates of your data on a three-dimensional graph: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$
- More independent variables extend this into higher dimensions
- The plane (or higher dimensional shape) will be placed so that it minimises the distance (sum of squared errors) to every data point

Linear regression

Advantages

- Simple
- Fast
- Interpretable
- Often surprisingly accurate

Disadvantages

- Can be too simple, linearity assumption overly strong

Break

Experimental Design

COMP20008

School of Computing and Information Systems

Experimental Design (supervised)

Evaluation methods

Performance metrics

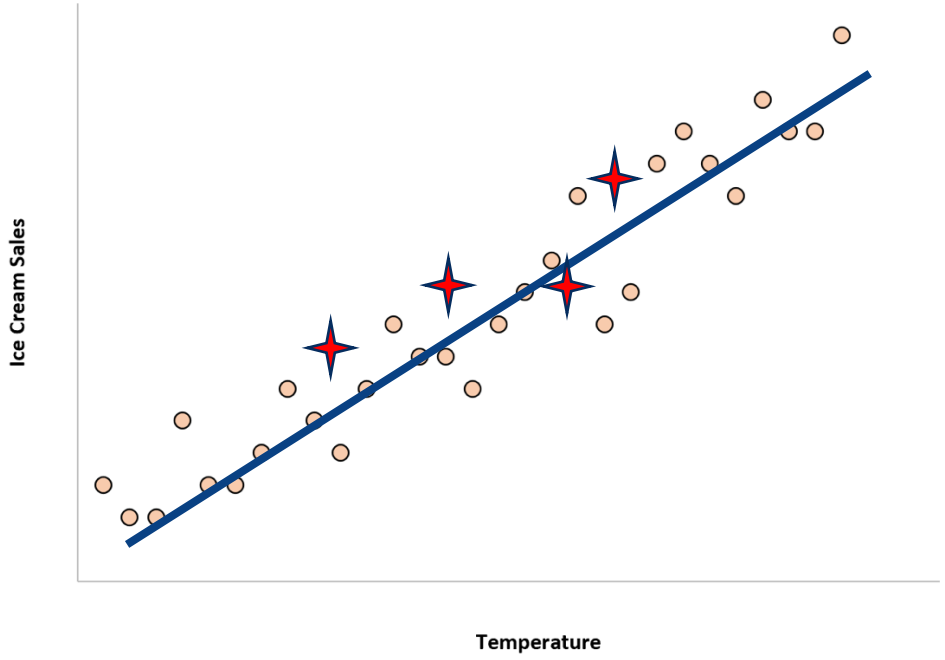
Feature selection

Supervised vs Unsupervised Learning

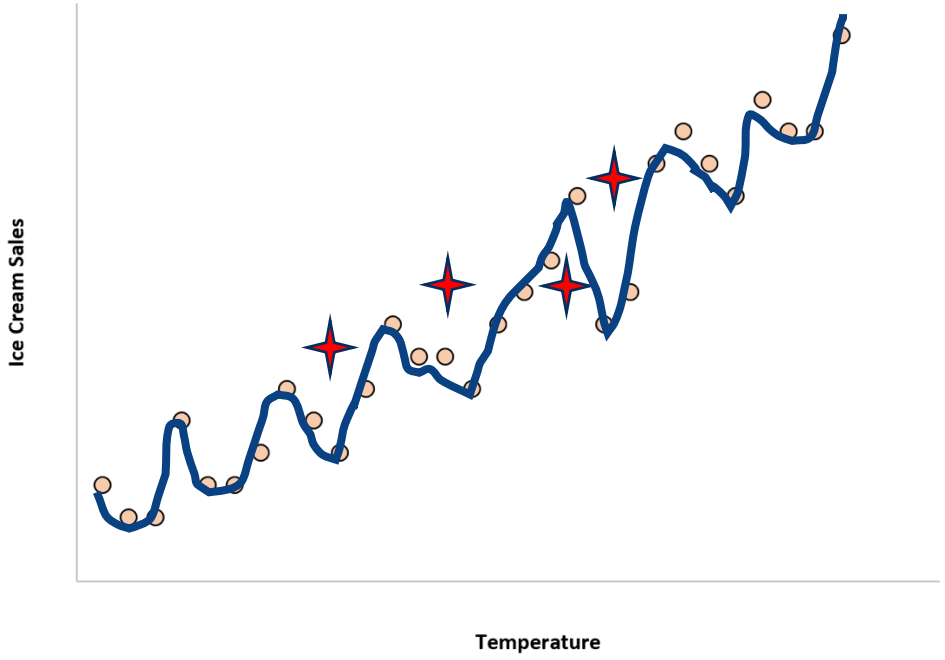
	Data	Model used for
Supervised learning	Labelled	Predict labels (categories or numbers) on new samples. E.g. k-nn or decision tree
Unsupervised learning	Unlabelled	Cluster related instances; E.g. Clustering, dimensionality reduction

The generalisation challenge

Model 1



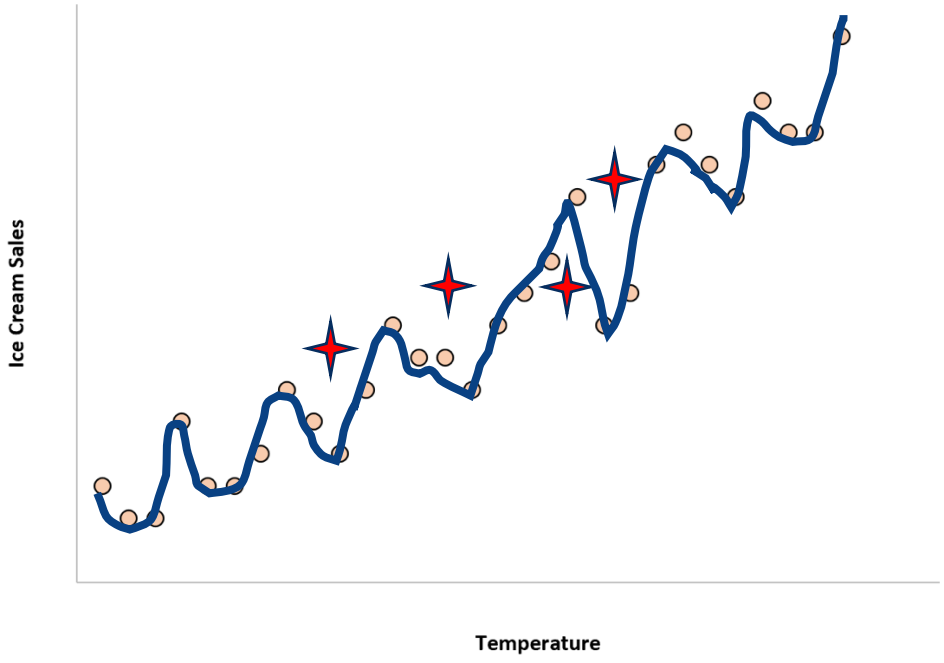
Model 2



✓ – Generalises well

✗ – Overfits the data
– Low Bias error
– High Variance error

The generalisation challenge – cont.



When a model learns too much from the training data: it has

– Low bias error
  - Predicts well on training data

– But high variance error
  - Predictions change significantly given different training data sets.

And it is overfitting –when it gives accurate results for the training set but does not generalise for the test set. Making an assumption that the past is exactly the same as the future.

Evaluation method – training and test sets

How do we know if our model will do well (e.g. have high accuracy) on unseen data?

We train the model on a set of data – the training set.

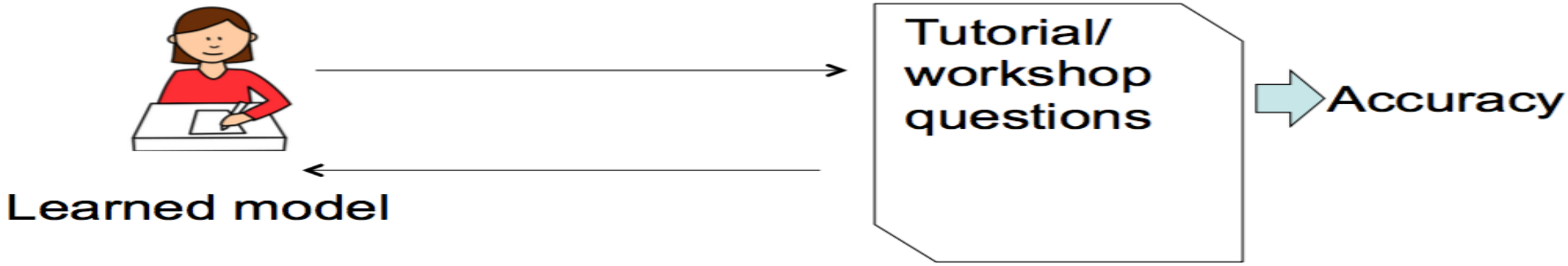
We evaluate the model on a new set of data – the test set.

Assumptions:

- Samples in both training and test sets come from the same distribution
- Samples are drawn independently from this distribution

Only one set? Partition the set into training set and test set!

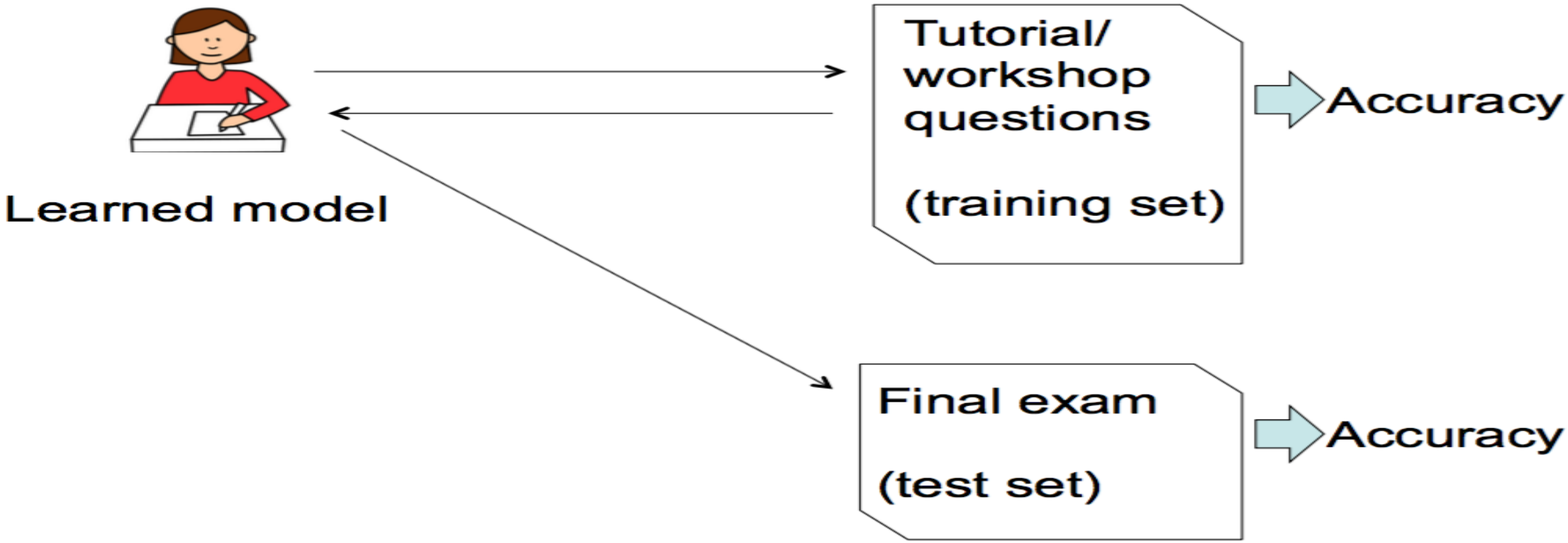
Why do we split the dataset into training and testing for evaluating accuracy?



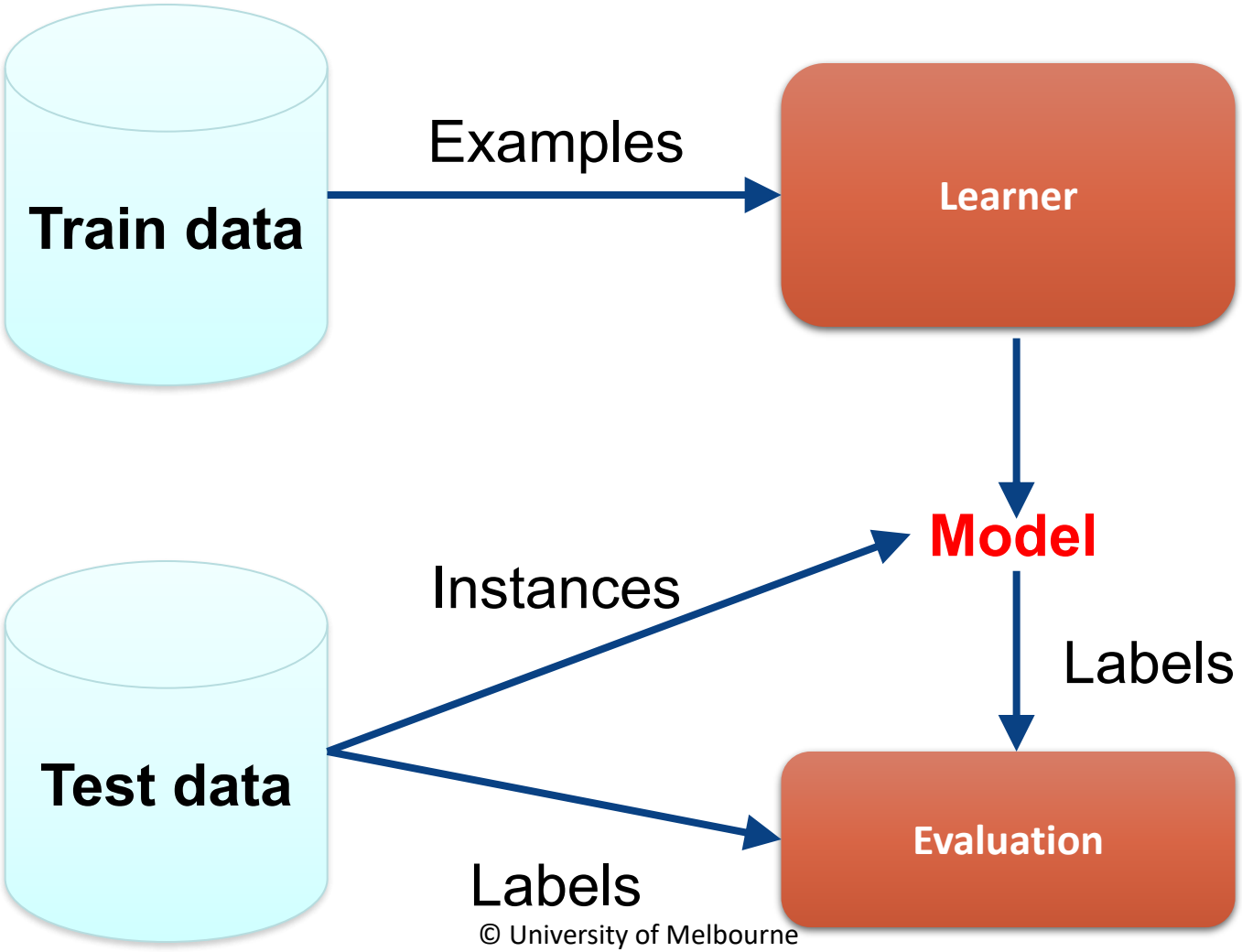
Learned model

Final exam

Accuracy



Architecture of a Supervised Learner



Evaluation (Supervised Learners)

How you measure quality depends on your problem!

Typical process

- Pick an evaluation metric comparing label vs prediction
- Procure an independent, labelled test set
- “Average” the evaluation metric over the test set

Example evaluation metrics

- Accuracy, Precision-Recall, Root Mean Squared Error, …

Cross-Validation

It is important to evaluate the performance on data not used for training

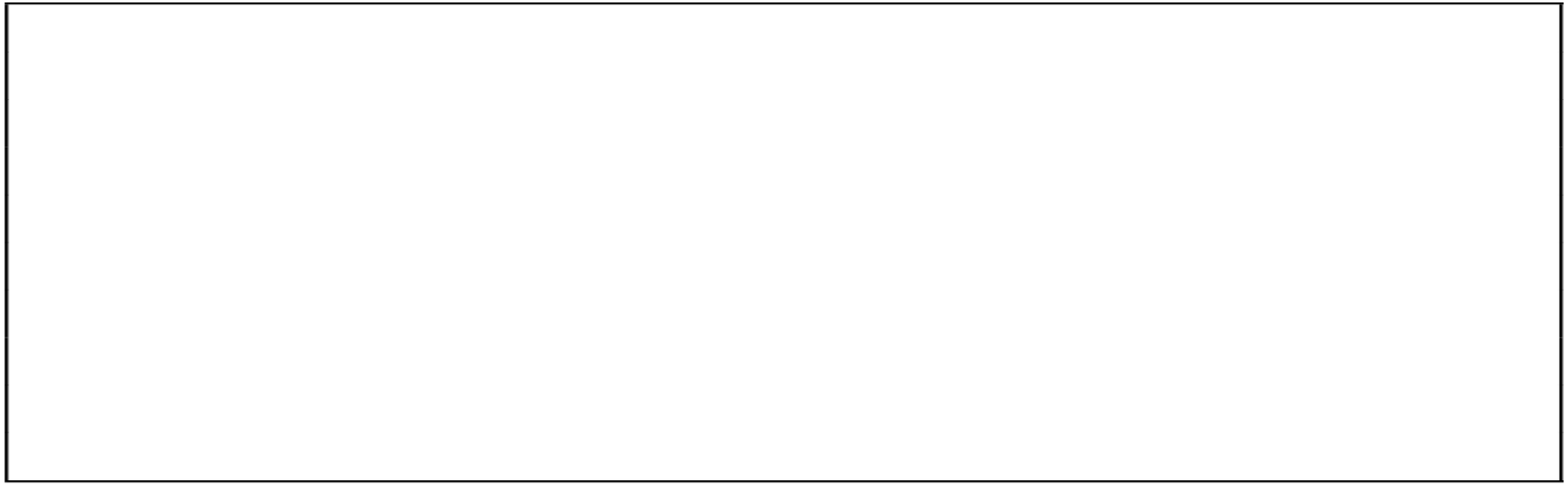
Hence, dividing the available data into train and test sets

When not much data available, cross-validate

In workshops: `from sklearn.model_selection import KFold`

Cross validation

- Take training data:



Cross validation (with N=10)

- Split up into N equal-sized partitions P_i:

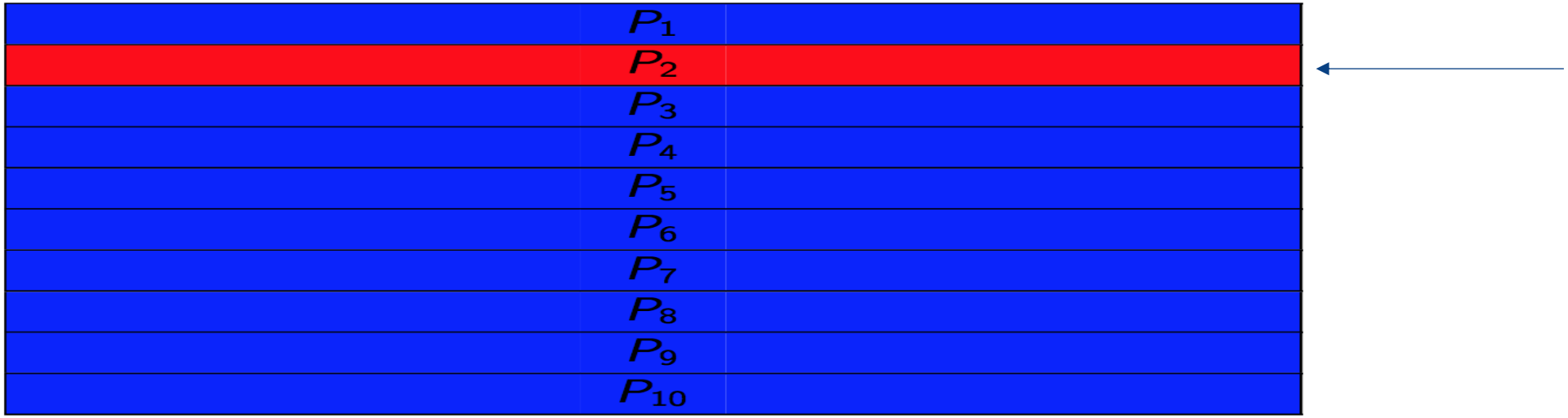
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_{10}

Cross validation

- For each $i = 1 \ldots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data

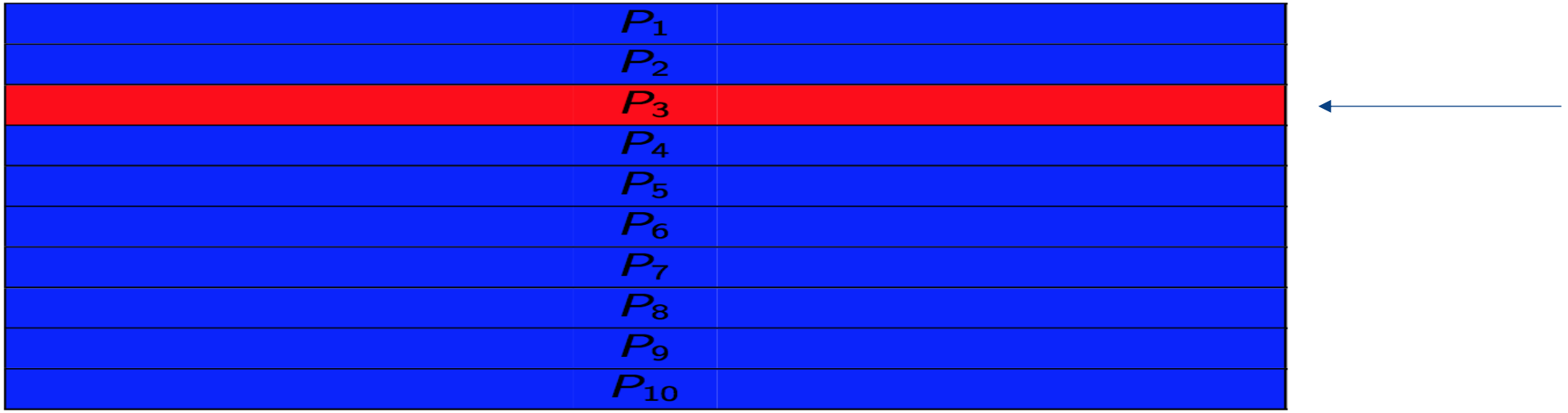
Cross validation

- For each $i = 1 \ldots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data



Cross validation

- For each $i = 1 \ldots N$, take P_i as the test data and $\{P_j : j \neq i\}$ as the training data



Then it will continue in each iteration to take test data then the rest training:
P_4 $P_{5..}$ P_{10}

Cross validation

The splits are made randomly

Generally N=10 partitions ("sweet spot" of N=10 found by Ron Kohavi, 1995), but sometimes 5 if dataset is small

- Average performance over the partitions (e.g. an average of 10 accuracies)

Optional: if you are curious why N = 10 in cross validation is commonly used:

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 (IJCAI'95). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1137–1143.

Stratified cross validation

It is common to stratify the data during partitioning

- The splits are made semi-randomly
- Each partition is created in a way that maintains the overall class distribution
  - e.g. If classes are High (80%) and Low (20%), then both training and test should have this 80:20 High:Low ratio.

Leave one out sampling

Extreme version of cross validation, where there is a single instance per partition

- 10,000 instances -> 10,000 partitions
- What is the advantage and disadvantage here?

How does cross validation work in practice?

Given: dataset (X, y), list of candidate k values for k-NN, number of folds (K)
Goal: Find the optimal k for k-NN

For each candidate k in [1, 3, 5, 7, 9, ...]:
  Initialise list of accuracies = []
    Split dataset into K folds (randomly, once at the beginning)

  For each fold i = 1 to K:
      Validation set = fold i
      Training set = all other folds
      Train k-NN model with current k on Training set
      Predict on Validation set
      Compute accuracy
      Append accuracy to accuracies list

  Compute mean_accuracy = average(accuracies)
  Store mean_accuracy for this k

Building and evaluating your model (includes hyperparameter selection)

Choose the k with the highest mean_accuracy
Retrain final k-NN model using chosen k on the FULL dataset (X, y)
Use this final model for future predictions

Building your final model

Repeated cross validation

Repeat N-fold cross validation r times, for example 5 times, to smooth effect of random selections

Procedure

1. Repeat r times:
   i. Partition dataset randomly into N blocks
   ii. Repeat N times:
   - *N-1* blocks for training and *1* block for testing
   - Record 1 performance score
2. Average the N× r scores

Poll EV Questions

URL: https://go.unimelb.edu.au/8b7p

Hyperparameter selection

Suppose we wish to choose the best k for a k-NN classification model and evaluate performance of resulting model

- Here k is usually referred to as a hyperparameter

We should not look at the test data, when choosing k! (avoid data leakage)

So we need to create a subset of the training set to act like the test set. We call this the validation set.

Training – validation – test split for k=1,3,5

Using the the validation set for hyperparameter optimization to prevent data leakage.

Repeat for k in [1,3,5]

- Fit a k-nn model on the Training-subset and record its performance on Validation

Select the k that performs best on Validation

Use merged (Training-subset + Validation) to fit the k-nn model using the selected k

Report its performance on the independent Test

Generating random training/test splits

Cross validation is a common way to manufacture multiple training/test splits

Another different approach is the use of **bootstrap sampling**, described next. It is a common method in statistics.

Evaluation method – Bootstrap validation

Bootstrap – A person pulling them self up and over a fence by pulling upwards on their own bootstraps.

https://en.wikipedia.org/wiki/Bootstrapping#/media/File:Dr_Martens,_black,_old.jpg

Relying on lots smaller samples to draw conclusions

Bootstrap sample – each smaller sample is drawn randomly from the original dataset using replacement. Each bootstrap sample is training data.

Out of bag data (OOB): remaining data points NOT in the bootstrap sample; these are the test data.

Bootstrap sampling

Method: construct "near-independent" datasets via sampling with replacement

- Generate N datasets, each size n sampled from n training samples with replacement – bootstrap samples
- Fit model on each constructed dataset
- Aggregate predictions via voting/averaging

Original training dataset:

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Bootstrap samples:

$\{7, 2, 6, 7, 5, 4, 8, 8, 1\ 0\}$ – out-of-sample/out-of-bag $3, 9$

$\{1, 3, 8, 0, 3, 5, 8, 0, 1, 9\}$ – out-of-sample/out-of-bag $2, 4, 6, 7$

$\{2, 9, 4, 2, 7, 9, 3, 0, 1, 0\}$ – out-of-sample /out-of-bag $5, 6, 8$

Bootstrap validation (bootstrapping) – cont.

Procedure

1. Draw b bootstrap samples (from training data)

   - Repeat b times, for each bootstrap sample:

   – Train the model on the bootstrap sample,

   – Evaluate the performance on its out of bag (OOB) (test/validation data)

   - Report the mean and standard deviation of the b performance scores

Can handle imbalanced data sets by using stratified bootstrap where biased sampling is used to maintain ratio of labels

Bootstrap cont.

Can show that for bootstrap sampling

- Training data: Each bootstrap sample contains ~63% of data
- Test data: Each out of bag sample contains ~37% of the data

Discussion question

- What might be the relative merits between using
  - 10-fold cross validation for performance assessment vs.
  - Bootstrap sampling with 1000 samples for performance assessment

Break

Experimental Design (supervised)

Evaluation methods

Performance metrics

Feature selection

Classification metrics

Confusion Matrix

The outcomes of the classification can be summarised in a Confusion Matrix (contingency table)

Actual class: {yes, no, yes, yes, …}

Predicted class: {no, yes, yes, no, …}

TP: true positive

FN: false negative

FP: false positive

TN: true negative

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	#TP	#FN
	Class=No	#FP	#TN

Classification metric – Accuracy

How many observations are correctly classified out of n observations

$$Accuracy = \frac{\#TP + \#TN}{n}$$

n is the total number of observations. $n = \#TP + \#FN + \#FP + \#TN$

$\frac{45 + 51}{45 + 3 + 1 + 51}$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	45	3
	Class=No	1	51

Accuracy

Accuracy is misleading in imbalanced problems.

$$\frac{97+0}{97+0+3+0} = 0.97$$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	97	0
	Class=No	3	0

The predictions for the minority class are completely wrong but the overall accuracy value is high.

Classification metric – Recall

Recall – Effectiveness of a classifier to identify class labels – true positive rate.

$$Recall = \frac{\#TP}{\#TP + \#FN}$$

$$\frac{45}{45 + 3} \approx 0.94$$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	45	3
	Class=No	1	51

Use Recall to minimise FN (for example, aim to detect as many malicious programs as possible)

Classification metric – Precision

Precision – Agreement of the true class labels with those of the classifier's

$$Precision = \frac{\#TP}{\#TP + \#FP}$$

$$\frac{45}{45 + 1} \approx 0.98$$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	45	3
	Class=No	1	51

Use precision when you DON'T want FP (avoid putting innocent people in prison)

Example – Rapid Antigen Test (RAT)

Recall (sensitivity): 85-95%

Precision: 86-93%

Classification metric – F1

F1 – The harmonic mean between precision and recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$2 \times \frac{0.98 \times 0.94}{0.98 + 0.94} \approx 0.96$$

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	45	3
	Class=No	1	51

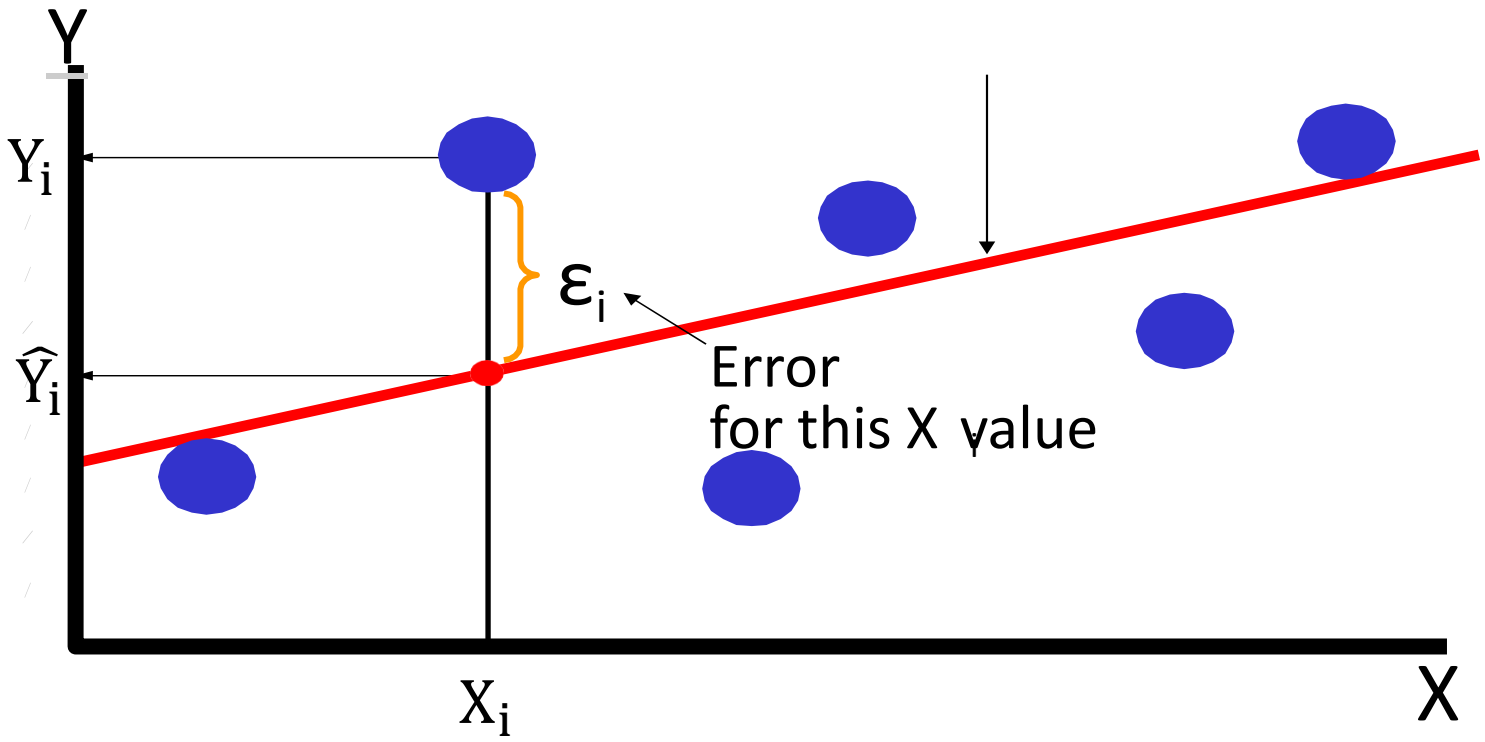
Regression metrics

Linear Regression

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Error term or the residual

$$\varepsilon_i = Y_i - \hat{Y}_i$$



Regression metrics

MSE – Mean Squared Error, SSE – sum of squared errors

Lower is better

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

$$SSE = \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

$$MSE = \frac{SSE}{N}$$

Regression metrics – cont.

RMSE – Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{N}}$$

$$= \sqrt{MSE}$$

$$= \sqrt{\frac{SSE}{N}}$$

Another commonly used measure

Lower value is better

Regression metrics – cont.

MAE –Mean Absolute Error

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \hat{Y}_i\right|$$

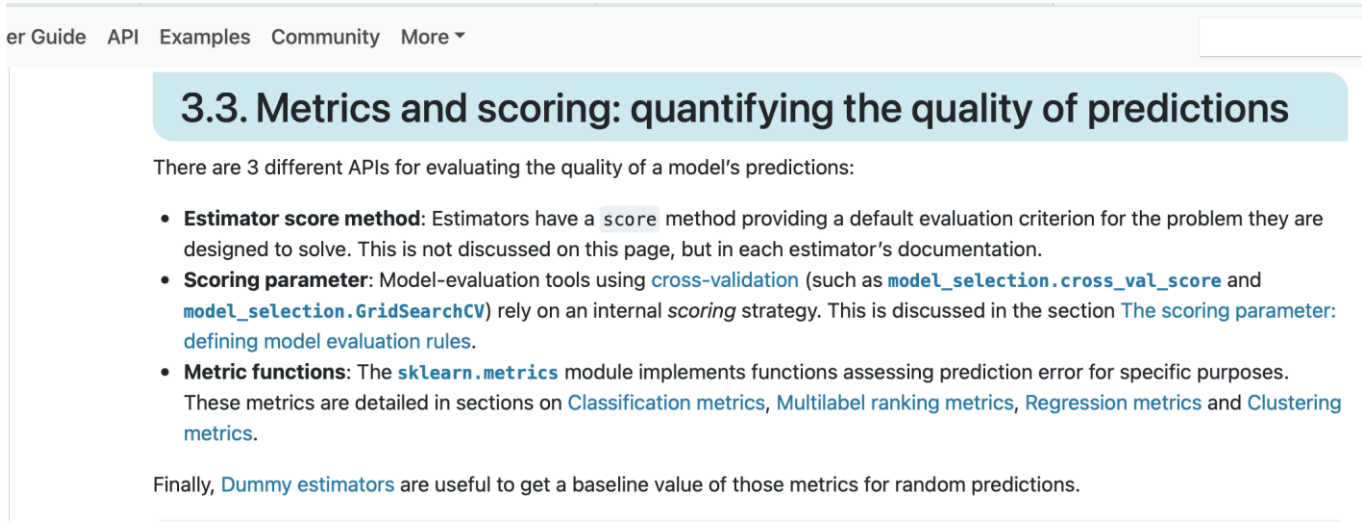
Lower value is better

More robust against outliers compared to MSE and RMSE

Others

Others (MAPE, Median Absolute Error)

https://scikit-learn.org/stable/modules/model_evaluation.html



Break

Experimental Design (supervised)

Evaluation methods

Performance metrics

Feature selection

Feature selection – univariate

Intuition: evaluate “goodness” of **each** feature

Consider each feature separately: one at a time

Typically, most popular and simple strategy for selecting features

Feature selection for classification

What makes a single feature good? **Well correlated with class**

- So likely to improve performance of a prediction model

Which of a_1, a_2 is a good feature for predicting the class c?

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

Predicting Whether Telstra Stock Rises

Today's Telstra share price.

Today's All Ordinaries Index.

Yesterday's Telstra share price.

The day before yesterday's Telstra share price.

Today's Optus share price.

Today's temperature in Melbourne.

Today's temperature in Sydney.

Today's temperature in Canberra.

Was there a front page story about Telstra in today's newspapers ?

The difference between Telstra's share price today and its share price one month ago.

The value of Telstra's share price multiplied by Optus' share price.

The colour of the eyes of the current Chief Executive Officer for Telstra.

Did yesterday's Chief Executive Officer for Telstra have different colour eyes from today's Chief Executive Officer for Telstra ?

Today's price for BHP shares.

Today's cheapest price for an airfare between Melbourne and Sydney.

Ranking-based Feature Selection

The simplest, most efficient form of filter-based feature selection is ranking, whereby we rank each feature X_i based on analysis of its predictiveness.

For discrete valued features, an algorithm we have seen already for doing this:

- *Mutual information*

The final set of features can be determined based on a threshold for predictiveness or taking the top-*N* features from the ranking

Feature Correlation: Mutual Information

$$\begin{aligned} MI(X,Y) &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y) \end{aligned}$$

MI is a measure of correlation

- the amount of information about *X* we gain by knowing *Y*, or
- the amount of information about *Y* we gain by knowing *X*

The amount of information shared between two variables *X* and *Y* *MI* (*X*,*Y*)

- large: *X* and *Y* are highly correlated (more dependent)
- small: *X* and *Y* have low correlation (more independent)
- $0 \leq MI(X,Y)$

To compute *MI*(*X*,*Class*)

- Class must be categorical (if it isn't, then discretise it into bins)
- *X* must be categorical (if it isn't, then discretise it into bins)

Feature selection – Mutual information (MI)

What makes a feature good? If it is well correlated with the class

Mutual Information

$$MI(X, Y) = H(Y) - H(Y|X)$$

Is feature a well correlated with the class c?

$$MI(a, c) = H(c) - H(c|a)$$

High $MI(a, c)$: a strongly predicts c; select a into the feature set

Low $MI(a, c)$: a can not predict c; a is not selected into the feature set

Feature selection – MI – cont.

Is a_1 well correlated with the class c?

$$MI(a_1, c) = H(c) - H(c|a_1)$$
$$= 1 - 0 = 1 \text{ (High MI, Yes!)}$$

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

The feature a_1 perfectly predicts c; select a_1 as a feature.

Is a_2 well correlated with the class c?

$$MI(a_2, c) = H(c) - H(c|a_2)$$
$$= 1 - 1 = 0 \text{ (Low MI, No!)}$$

a_1	a_2	c
Y	Y	Y
Y	N	Y
N	Y	N
N	N	N

The feature a_2 can not predict c at all; a_2 is Not selected as a feature.

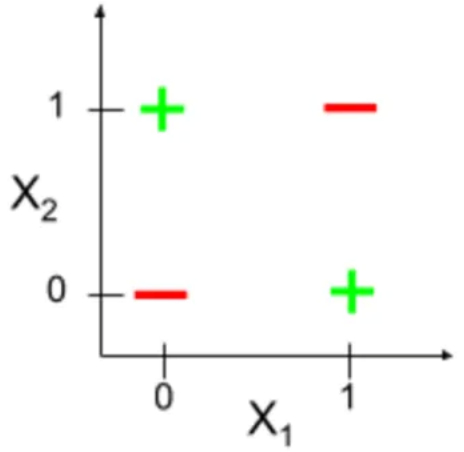
Univariate feature selection – potential issues

Difficult to control for inter-dependence of features

Feature filtering of single features may remove important features. For example, where the class is the XOR of some features:

- Given all the features, the class is totally predictable.
- Given one of them, the MI is 0.

In practice, feature extraction may be used, i.e. construct new features out of existing ones, e.g. ratio / difference between features



Income	Expenditure	i>e	Credit
120	100	1	Good
50	30	1	Good
50	70	0	Bad
200	40	1	Good
200	210	0	Bad
…	…	…	…
160	150	1	Good

Feature selection for regression

1. Mutual Information (the dependent variable (class) will need to be discretised)
2. Pearson Correlation

Model evaluation with feature selection (cross validation)

Feature selection should be done within each training step.

A set of selected features is like a hyperparameter

N-fold cross validation procedure

1. Partition *training data* randomly into N blocks
2. – Repeat N times:
   - Feature selection on the N−1 blocks,
   - N−1 blocks for training using selected features,
   - 1 block for evaluation
   – Average the N scores

Model evaluation with feature selection (bootstrap)

calculating features in each fold, prevent data leakage

Feature selection should be done within each training step.

Bootstrap validation procedure

1. Draw b bootstrap samples
2. - Repeat b times, for each bootstrap sample:
   - Feature selection on the *bootstrap sample*,
   - Train the model on the bootstrap sample using the selected features,
   - Evaluate the performance on the OOB (test/validation data)

   - Report the mean and standard deviation of the b performance scores

The overall flow with feature selection

1. Model evaluation and selection with feature selection
2. Apply feature selection and fit the selected model and hyper-parameters with the entire training set
3. Report performance on the independent test set.

Discussion question (past exam)

-Suppose Alice takes a dataset D with 100 samples, 4 features, plus a class label feature.

- First, she computes the correlation of each of the 4 features with the class label using mutual information and discards the two features with lowest correlation.
- Then, she now has a processed version D' of the dataset (2 features, class label feature and 100 instances).
- Afterwards, she splits D' into two -80% training (80 instances) and 20% testing (20 instances).
- Then, she fits a 3-NN model on the training set and evaluates the model accuracy on the testing set.
- After this process, she reports the accuracy as being 90%. Why might this estimate of 90% accuracy be over-optimistic? Give reasons.

he computed mutual information on all 100 rows (including those later used for testing) and dropped 2 features before the split. The test labels influenced which features were kept, so the model indirectly "saw" the test set.

End