Machine Learning
Introduction to ML Model Evaluation

IIIT Sri City

This week's Agenda

- Recap to classification?
- How to evaluate the classification model?
- Evaluation Metrics for a Classification model
- Recap to regression problem?
- How to evaluate the regression model?
- Evaluation Metrics for a regression model
- Dataset: Train, Validation and Test sets
- Train, test and validation split
- Data Sampling Methods
- Overfit and Underfit

Introduction to parametric and non-parametric machine learning models.

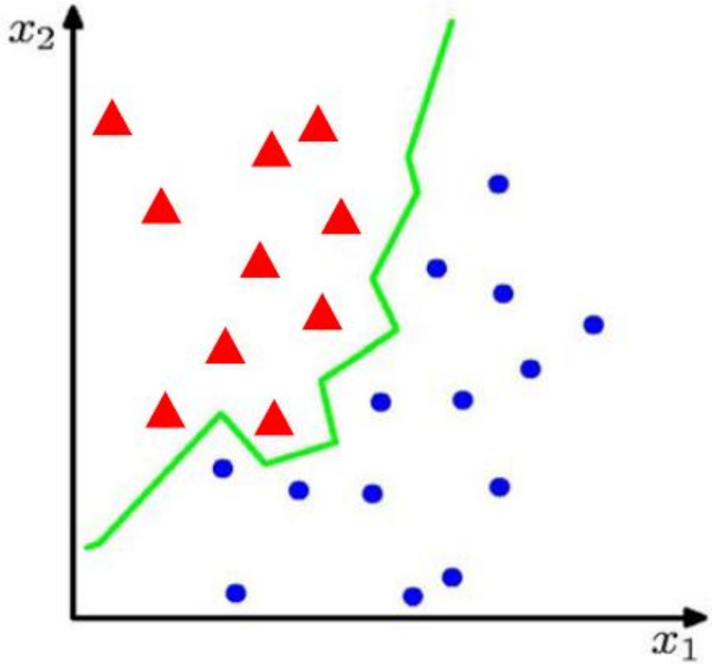
- *A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model.*
  - *No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs.*

- Some more examples of parametric machine learning algorithms include:
  - Logistic Regression
  - Linear Discriminant Analysis
  - Perceptron

- *Algorithms that do not make strong assumptions about the form of the mapping function are called nonparametric machine learning algorithms.*
  - *Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features.*

- Some more examples of popular nonparametric machine learning algorithms are:
  - k-Nearest Neighbors
  - Decision Trees like CART and C4.5
  - Support Vector Machines

Recap: What is classification problem?



- Suppose we are given a training set of N observations

$$(x_1, \ldots, x_N) \text{ and } (y_1, \ldots, y_N), x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$$

- Classification problem is to estimate f(x) from this data such that

$$f(x_i) = y_i$$

Classification: Supervised Learning

Training Phase

We have shown a set of dog pictures and a set of cat pictures to a child.

Classification: Supervised Learning

Testing Phase

→

→ DOG

This picture as it is
may not be in the
training set

Child has done more than just remembering

Evaluating a classifier

Classification Accuracy and Classification Error

Let the test set size be *n*.

Classification Accuracy (CA):

$$CA = \frac{\text{Number of correct predictions}}{N}$$

$$Error = \frac{\text{Number of wrong predictions}}{N}$$

CA and Error

In the previous formula we assumed that the test examples are all equally likely.

In other words, the Probability of drawing any of the test example is the same.

However, this need not be the case always.

Recap: Probability

Probability theory is built on 3 axioms.

The terms one should be clear:

1. Experiment
2. Sample space
3. Event
4. Probability

Kolmogorov Axioms of Probability

To the probabilities of outcomes/events of an experiment must obey the axioms:

- **Axiom 1**: For any event A, $\Pr(A) \geq 0$
- **Axiom 2**: $\Pr(\Omega) = 1$
- **Axiom 3**: For a collection of <u>*mutually exclusive*</u> events, $A_1, A_2, \ldots, A_n$

$$\Pr(A_1 \cup A_2 \ldots \cup A_n) = \sum_{i=1}^{n} \Pr(A_i)$$

- Everything else in probability theory can be deduced starting with these axioms

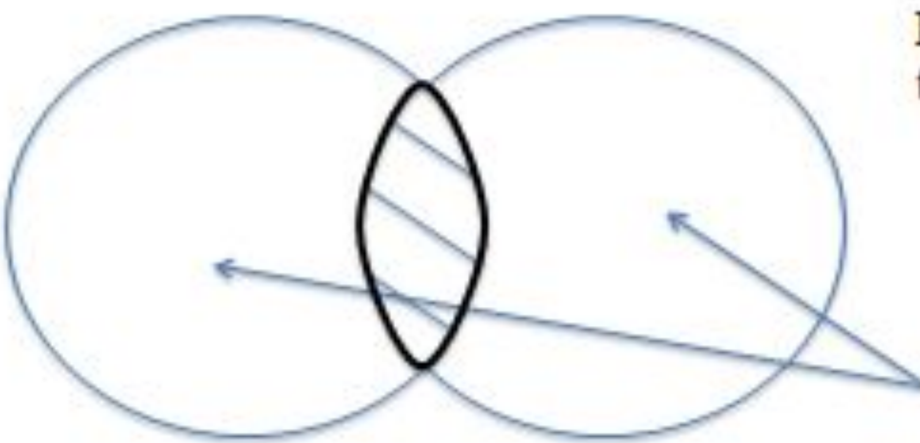
Handy Consequences of Kolmogorov Axioms

- <u>Important consequences</u>:
  - Probability of a union of non-disjoint events

$$\boxed{\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)}$$

<u>In words</u>: The probability of A *<u>or</u>* B is the probability of A *plus* the probability of B *minus* the probability of A *<u>and</u>* B

Don't count the probabilities of A and B twice if there is overlap between the events



$$\Pr(A \cup B)$$

Loss function

Let X be the given example with y being the target label (ground-truth).
Let the classifier predicted y'

Loss of this prediction, P(error|X) and P(error) :

$$\mathcal{L}(x) = \begin{cases} 0, & \text{if } y = y' \\ 1, & \text{otherwise} \end{cases}$$

$$P(\text{error}|x) = \mathcal{L}(x)$$

{This assumes that the feat. space is a discrete one}

$$P(\text{error}) = E[P(\text{error}|x)]$$

$$= \sum_{\forall x} \mathcal{L}(x)\, P(x)$$

If the space is continuous,

$$p(\text{error}|x) = \mathcal{L}(x)$$

$$p(\text{error}) = E\left[p(\text{error}|x)\right]$$

$$= \int \mathcal{L}(x)\, p(x)\, dx$$

{The integration is over the feat. space}

Example (discrete case)

Data point	Ground truth	Predicted label	Probability
X1	Cat	Cat	1/8
X2	Cat	Dog	1/4
X3	Dog	Dog	1/4
X4	Dog	Cat	3/8

P(error) = Average Error rate = $0(\frac{1}{8}) + 1(\frac{1}{4}) + 0(\frac{1}{4}) + 1(\frac{3}{8}) = \frac{5}{8}$.

Note: The entire space is covered by these four examples.

Eg 2

Assume we are working with 1D, two class problem. Let the class labels are 1 and 2.

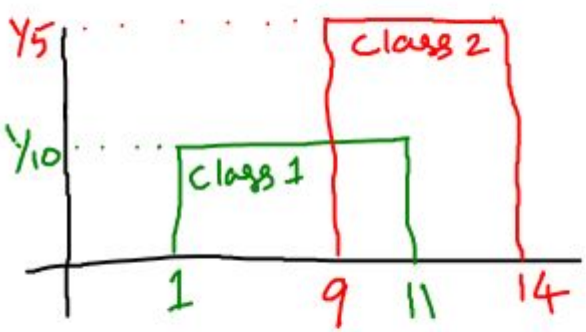
Let the classifier is a simple rule: if the feature value (x) is less-than 10 the class assigned is 1, else the class assigned is 2.

$$x < 10 \Rightarrow class = 1$$

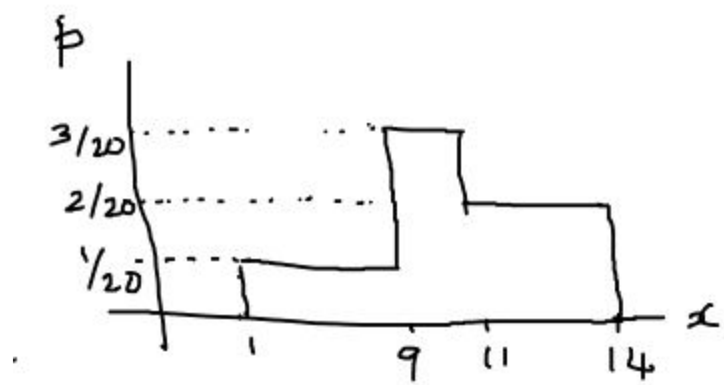
$$x \geq 10 \Rightarrow class = 2$$

Eg 2 Continued

Assume that the data is drawn from the mixture of the two class-conditionals (shown below) where each class is equally likely.

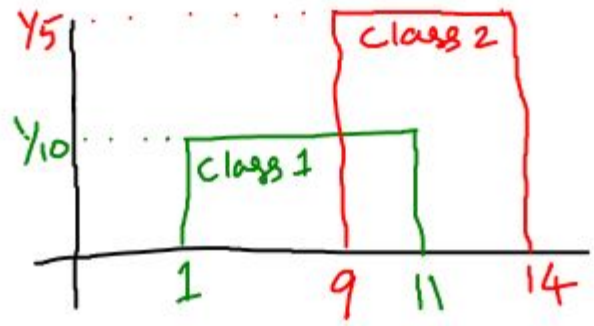


Class-conditional distributions



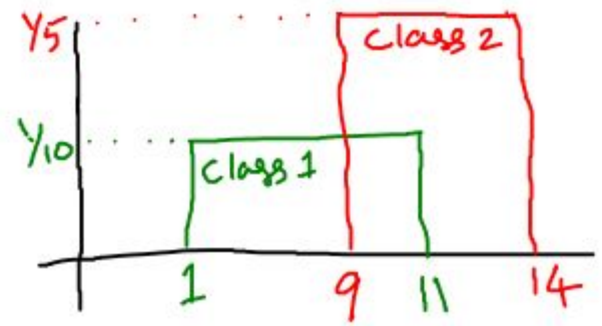
Distribution from which x is drawn

Eg 2



The classifier is not making any mistake for x < 9 and for x > 11. Only when x is in [9, 11] one has to worry about the mistakes made.

Eg 2

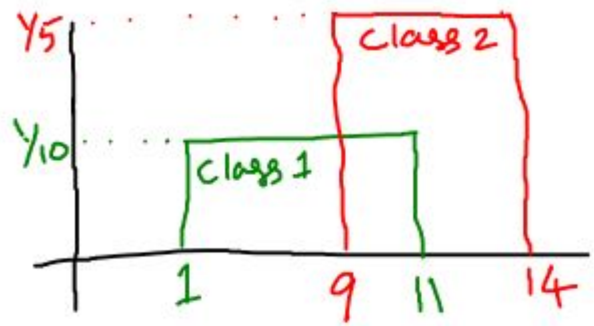


[9, 10] error: Let X is an arbitrary element drawn in [9,10]

P(err | X) = P(y=class2|X)

We use the notation P(class2|X) instead of P(y=class2|X)

Eg 2



[9, 10] error: Let X is an arbitrary element drawn in [9,10]

$$P(err \mid X) = P(class2 \mid X)$$

$$P(class2 \mid X) = \frac{p(X, class2)}{p(X)}$$

$$= \frac{p(X \mid class2)\, P(class2)}{p(X)}$$

$$p(X) = p(X|class1)P(class1) + p(X|class2)P(class2)$$

So, for X in [9,10],

$$P(class2|X) = \frac{\frac{1}{5} \cdot \frac{1}{2}}{\frac{1}{5} \cdot \frac{1}{2} + (1/10) \cdot \frac{1}{2}} = \frac{2}{3}$$

[10,11] error: Let X is an arbitrary element drawn in [10,11]

$$P(err \mid X) = P(class1 \mid X)$$
$$= \tfrac{1}{3}$$

P(error)

$$P(err) = \int P(err|x)\, p(x)\, dx$$

$$= \int_1^9 P(class\,2|x)\, p(x)\, dx$$

$$+ \int_9^{10} P(class\,2|x)\, p(x)\, dx$$

$$+ \int_{10}^{11} P(class\,1|x)\, p(x)\, dx$$

$$+ \int_{11}^{14} P(class\,1|x)\, p(x)\, dx$$

$P(error) = 0 + \frac{1}{5}.\frac{1}{2} + 1/10.(\frac{1}{2}) + 0 = 3/20$

Eg2 Contd

By shifting the threshold of the classifier can we reduce the error rate of the classifier?

If so, what would be the best possible threshold?

This actually points to the Bayes Classifier.

Actual error-rate

The actual error-rate calculation is theoretical and is possible when the distributions (Probability structure) of the given problem is known.

In reality we will not be having this.

Test set

Test set is an independently drawn set from the same distribution as the training set.

Since we are given with an example set, normal practice is to divide this into **the training set** and **the test set**.

Note that this division has to be random and in practice often disjoint division is followed.

Multiple test sets are possible

Since multiple training and test set divisions are possible, often we measure the accuracy or error as the average one over multiple divisions.

This can give us variance or standard deviation of the computed accuracy or error.

Later we will see about **bias and variance** of the learning.

Other evaluation metrics (especially for binary classification)

Total number of Dog Images (10), Cat (10), N=20

Confusion Matrix: Normally this matrix is of size *K*K* where *K* is number of classes.

		Predicted	
	N=20	Dog (+)	Cat (-)
Actual	Dog (+)	7	3
	Cat (-)	4	6

		Predicted	
	N=20	Dog (+)	Cat (-)
Actual	Dog (+)	TP	FN
	Cat (-)	FP	TN

True Positives: 7 (Dogs images were classified as Dog)

False Positives (*Type 1 Error*): 4 (Cats images were classified as Dog)

True Negatives: 6 (Cats images were classified as cat)

False Negatives (Type 2 Error): 3 (Dogs images were classified as cat)

Total (N) =TP+FP+FN+TN=20

Evaluation Metrics for a classification model

- **Accuracy**: Accuracy is number of correct predictions out of total records.

  Accuracy=(*TP*+*TN*)/Total=13/20=65%.

- Misclassification rate or error rate:

  Error rate=(*FP*+*FN*)/Total=7/20=35%.

Accuracy Paradox:

Consider, Total number of Dog Images (19), Cat (1), N=20

		Predicted	
	N=20	Dog (+)	Cat (-)
Actual	Dog (+)	19	0
	Cat (-)	1	0

		Predicted	
	N=20	Dog (+)	Cat (-)
Actual	Dog (+)	TP	FN
	Cat (-)	FP	TN

Accuracy= (TP+TN)/Total=(19+0)/20=99%

Evaluation Metrics for a Classification model

- Precision (positive predicted value): Is the fraction of documents retrieved that are relevant.
  - It is the number of positive predictions divided by the total number of positive class values predicted.

    Precision=*TP*/(TP+FP)=19/(19+1)=19/20=99%.
  - Your classifier said something is positive, how much precise this decision is?
- Recall (sensitivity or true positive rate): It is the fraction of relevant documents that are retrieved.
  - It is the number of positive predictions divided by the number of positive class values in the test data.

    Recall=TP/(TP+FN)=19/(19+0)=100%.
  - Fractions of positives that are actually said to be positive by your classifier.

PRECISION, RECALL ARE USED IN INFORMATION RETRIEVAL SYSTEMS. These can be used to evaluate a binary classification.

F-1 Measure

- F-1 Measure: A balanced measure between precision and recall.
  - F-1 Measure=2*(Precision * Recall)/(Precision + Recall)
  - F-1=2*(0.99*1.0)/(0.99+1.0)=2*(0.99)/(1.99)=1.98/1.99=0.99

Evaluation Metrics for a Classification model

- Specificity (True Negative Rate): How often the ML model predicts negative samples correctly out of total negative samples.

  Specificity=TN/(TN+FP)=0/(0+1)=0%

  -Model has 0% effective in predicting negative samples as negative.

- False positive rate (FPR): How often the ML model predict the negative samples (cat) as positive (dog)?

  FPR=FP/(TN+FP)=1/(0+1)=100%

  Note: It can also be calculated as FPR=1-Specificity=1-0=100%

  ML model has 100% FPR, which means that every time model will classifies every Negative (Cat) sample as Positive (Dog).

- *Sensitivity and Specificity are most important in medical diagnosis.*

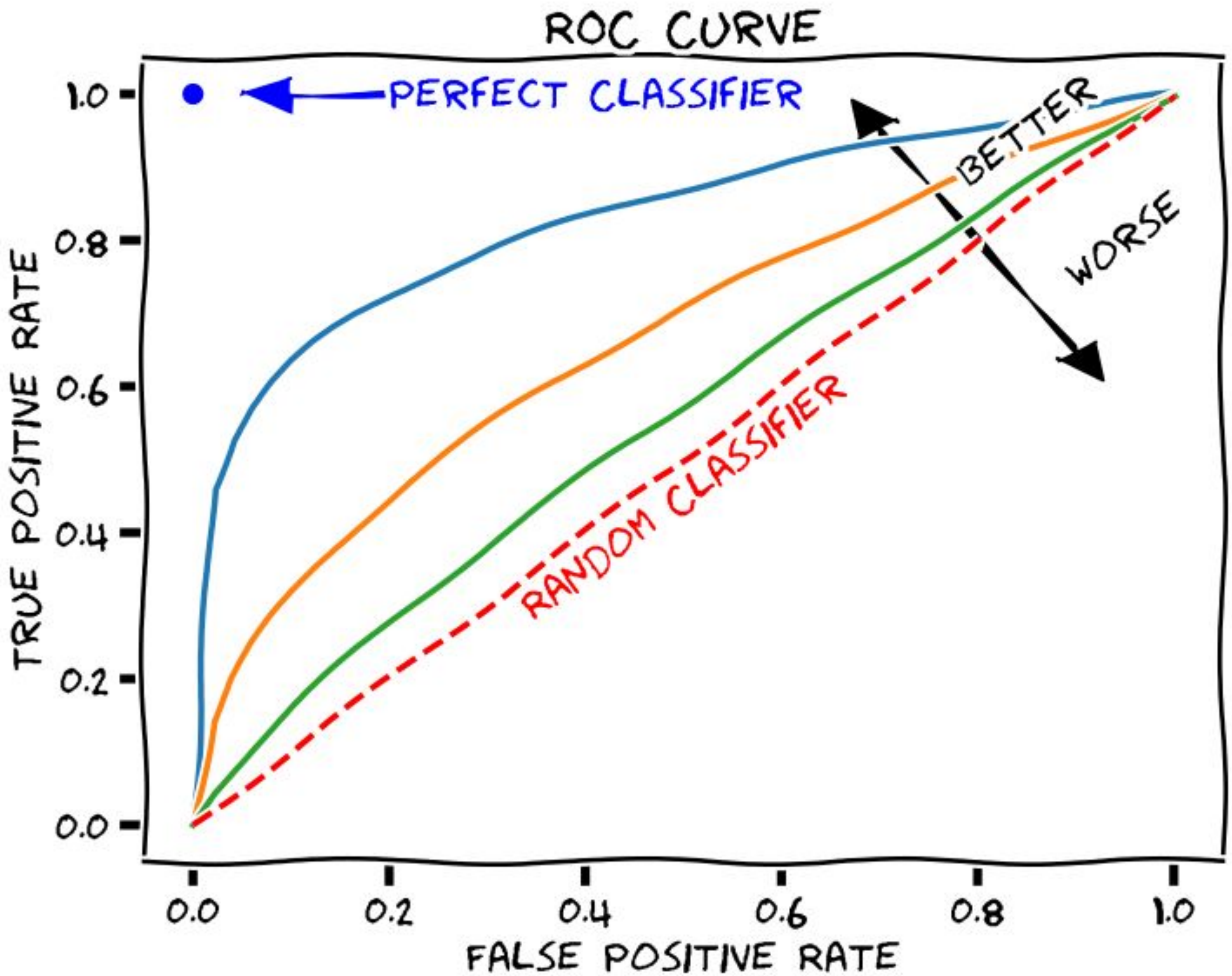
ROC Curve

- A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

- The method was originally developed for operators of military radar receivers, which is why it is so named.

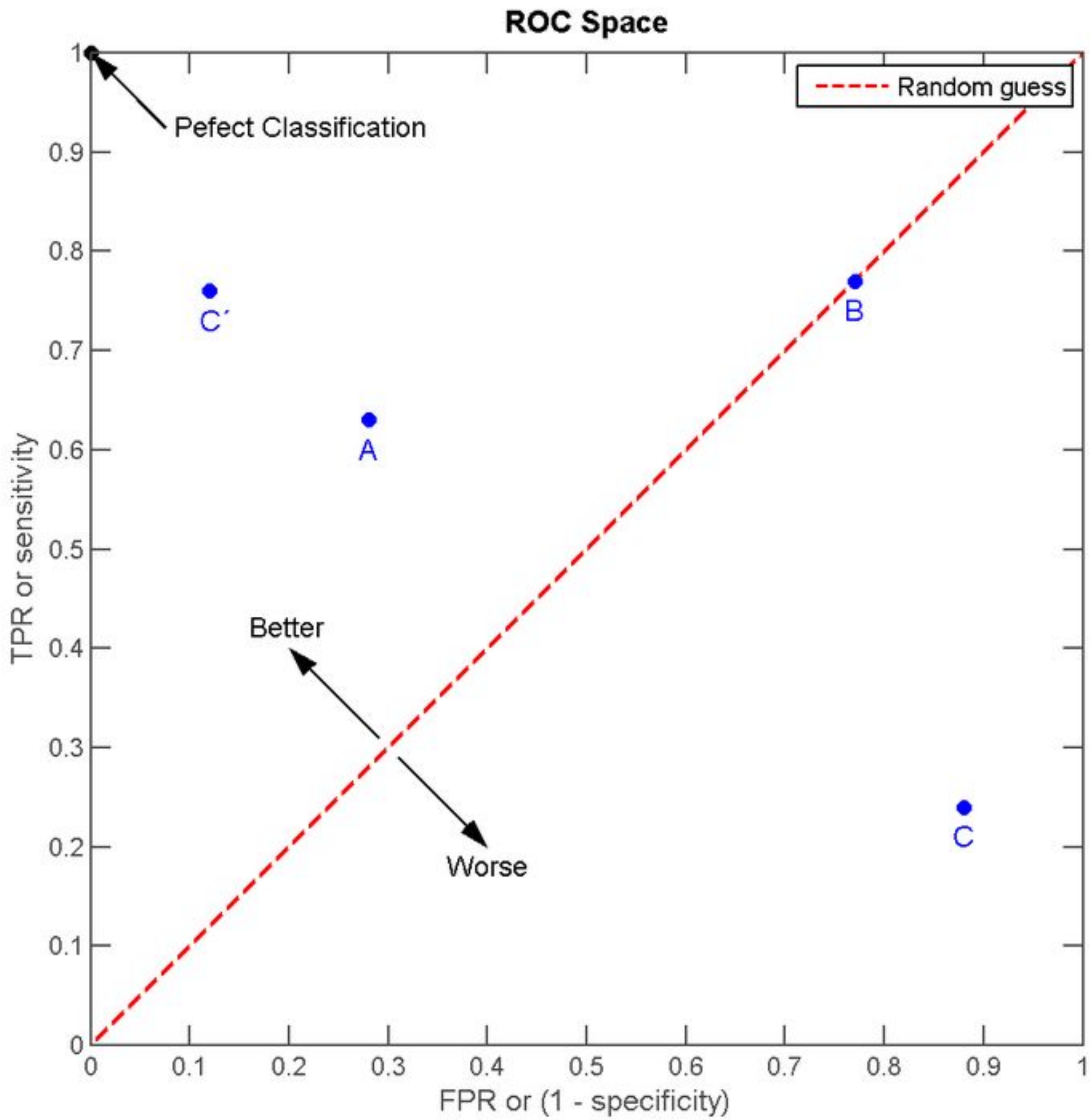
- The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

ROC Curve



ROC CURVE
PERFECT CLASSIFIER
BETTER
WORSE
RANDOM CLASSIFIER
TRUE POSITIVE RATE
FALSE POSITIVE RATE
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

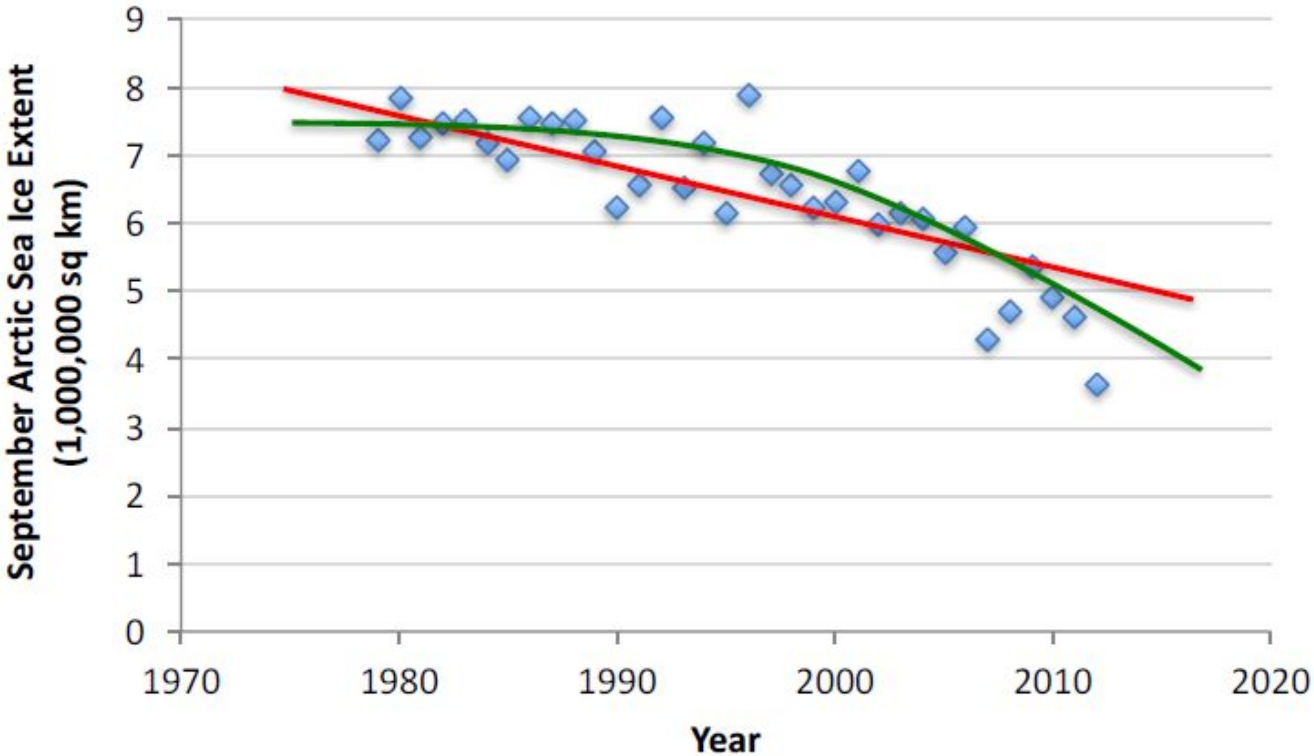
ROC Curve


ROC Space
Random guess
Pefect Classification
C´
A
B
C
Better
Worse
TPR or sensitivity
FPR or (1 - specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Regression

What is the Regression Problem?

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression

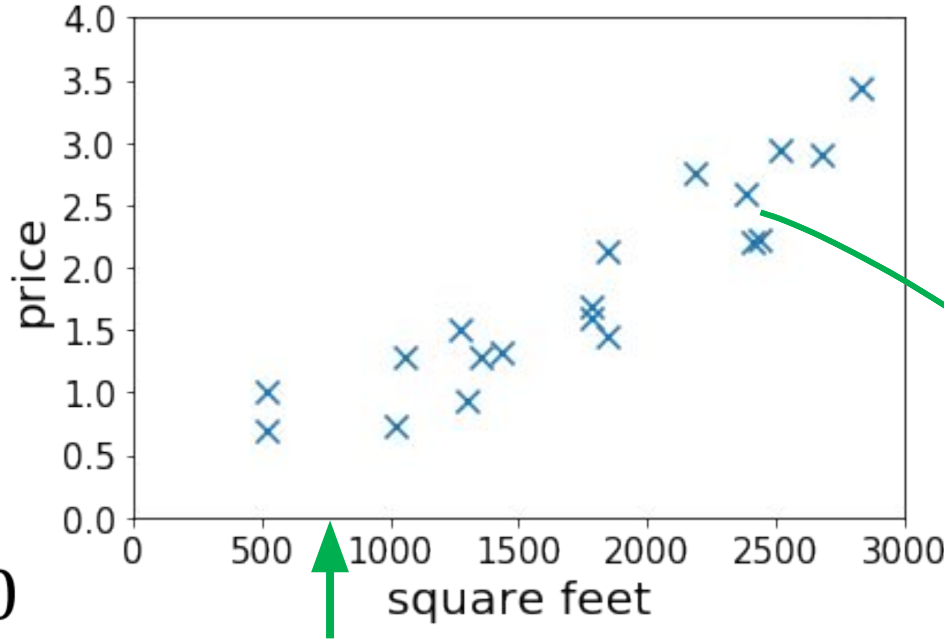


Housing Price Prediction

- Given: a dataset that contains n samples

$$\left(x^{(1)}, y^{(1)}\right), \dots \left(x^{(n)}, y^{(n)}\right)$$

- Task: If a residence has x square feet, predict its price?



15th sample $\left(x^{(15)}, y^{(15)}\right)$

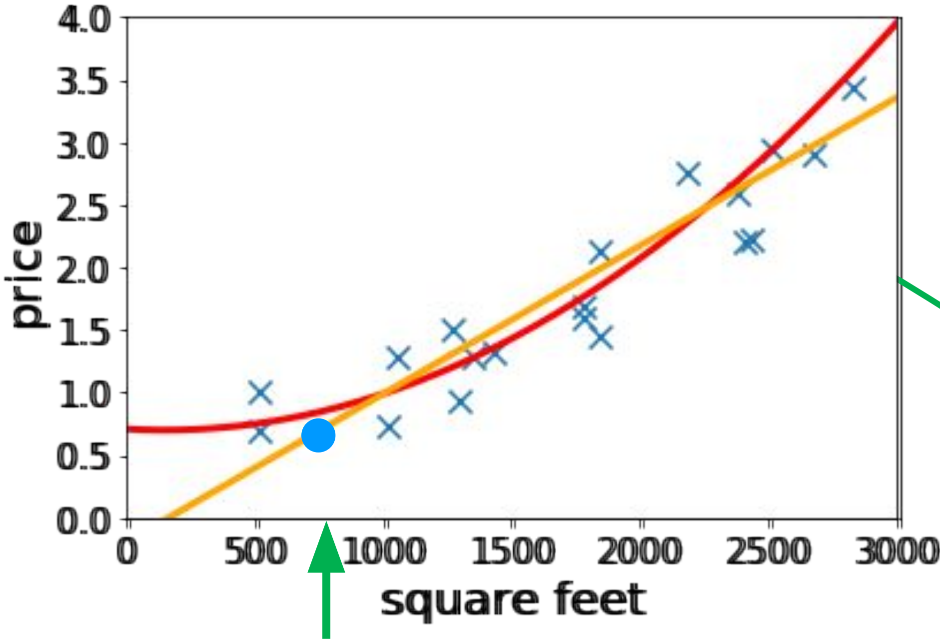
$$x = 800$$
$$y = ?$$

Housing Price Prediction

- Given: a dataset that contains n samples

$$(x^{(1)}, y^{(1)}), \ldots (x^{(n)}, y^{(n)})$$

- Task: If a residence has x square feet, predict its price?



$x = 800$

$y = ?$

15th sample

$(x^{(15)}, y^{(15)})$

- Solution: fitting linear/quadratic functions to the dataset.

How to evaluate the regression model?

- What is the accuracy of your model prediction?
- Compared to classification (where you have discrete set of labels as prediction) predicting the accuracy in regression (where you have continuous real value number) is slightly difficult.!
- It might be impossible for your ML model to predict the exact value.
- So to calculate the accuracy, you can compared the predicted value as how close it is against the real value.

Evaluation Metrics for a Regression model

- *There are three main metrics used for evaluating the regression models:*

  - *R Square/Square of the Correlation Coefficient*

  - *Mean Square Error(MSE)/Root Mean Square Error(RMSE)*

  - *Mean Absolute Error(MAE)*

Evaluating a regression model: R square value

- *R Square/coefficient of determination:* It measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

- *R* Square value is between 0 to 1 and bigger value indicates a better fit between prediction and actual value

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Where y_i is the original value, $\hat{y}_i$ is the predicted value and $\bar{y}$ is the mean of original values.

Evaluating a regression model: Mean Squared Error (MSE)

- MSE is calculated by the sum of square of prediction error which is real output minus predicted output and then divide by the number of data points.
- It gives you an absolute number on how much your predicted results deviate from the actual number.
- Root Mean Square Error(RMSE) is the square root of MSE.

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

Where y_i is the original value and $\hat{y}_i$ is the predicted value.

Evaluating a regression model: Mean Absolute Error (MAE)

- Mean Absolute Error(MAE) is similar to Mean Square Error(MSE).
- Unlike MSE where we take the sum of square of errors, MAE computes the sum of absolute value of error.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Where y_i is the original value and $\hat{y}_i$ is the predicted value.

- Note: MSE gives larger penalisation to big prediction error by square it while MAE treats all errors the same.

Dataset: Train, Validation and Test sets

- **Train set:** The model is initially fit on a training dataset,[3] which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model.

- In practice, the training dataset often consists of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), where the answer key is commonly denoted as the target (or label).

Dataset: Train, Validation and Test sets

- **Validation set:** Successively, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset.

- The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters e.g. the number of hidden units (layers and layer widths) in a neural network.

- The validation dataset functions as a hybrid: it is training data used for testing, but neither as part of the low-level training nor as part of the final testing.

Dataset: Train, Validation and Test sets

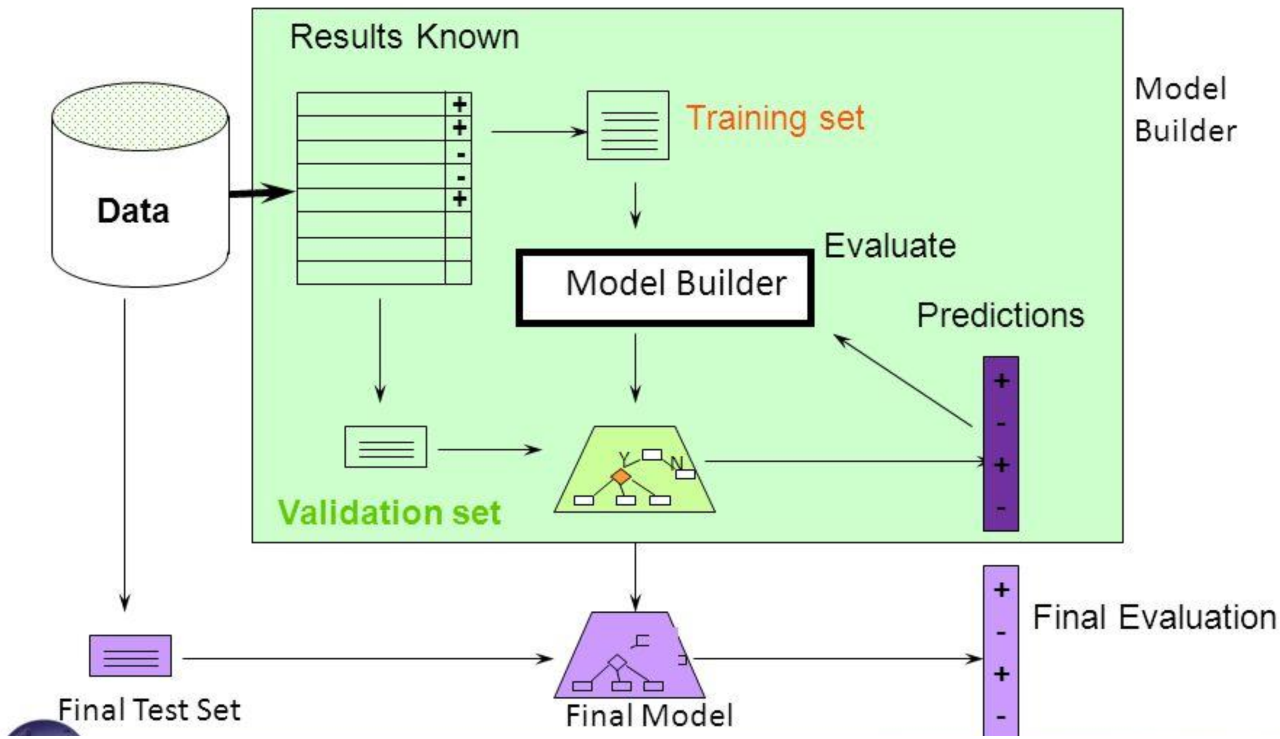
- **Test set:** Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset.

- The test dataset is typically used to assess the final model that is selected during the validation process.

- It is only used once a model is completely trained(using the train and validation sets).
  - Many a times the validation set is used as the test set, but it is not good practice.

- The test set generally contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

Train, test and validation split



Data Sampling Methods

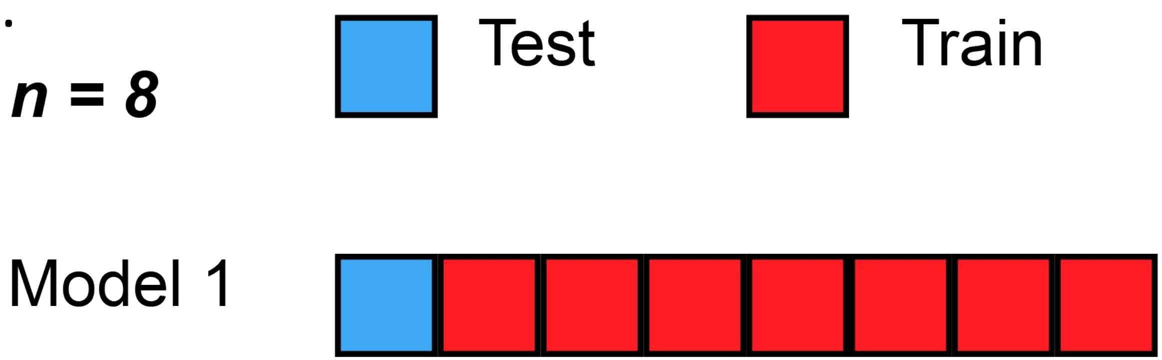
- **Hold out method**: In the holdout method, we randomly assign data points to two sets d0 and d1, usually called the training set and the test set, respectively.

- The size of each of the sets is arbitrary although typically the test set is smaller than the training set.

Data Sampling Methods

- **Cross Validation Method**: Cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (Validation set).

- To reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance.



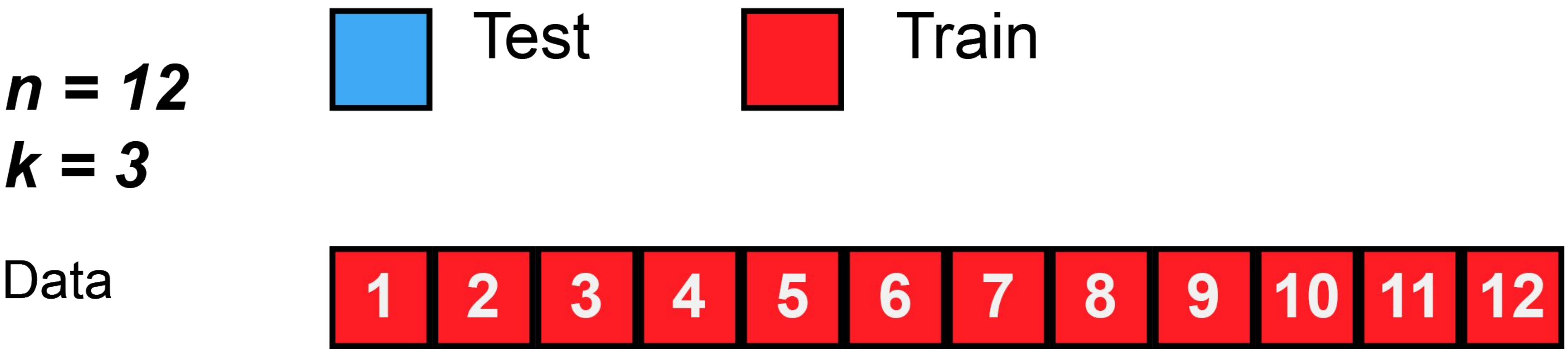
Data Sampling Methods

k-fold Cross validation: This procedure has a single parameter called “k” that refers to the number of groups that a given data sample is to be split into.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

k-fold Cross validation:



Overfit vs Underfit

- **Overfitting:** refers to a model that models the training data too well.
- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
- In overfitting, ML model learns the inference rules based on the noise or random fluctuations in the training data.

Overfit vs Underfit

- **Underfitting:** refers to a model that can neither model the training data nor generalize to new data.

- An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

- Underfitting is often not discussed as it is easy to detect given a good performance metric. The remedy is to move on and try alternate machine learning algorithms.

Thank You:
Question?