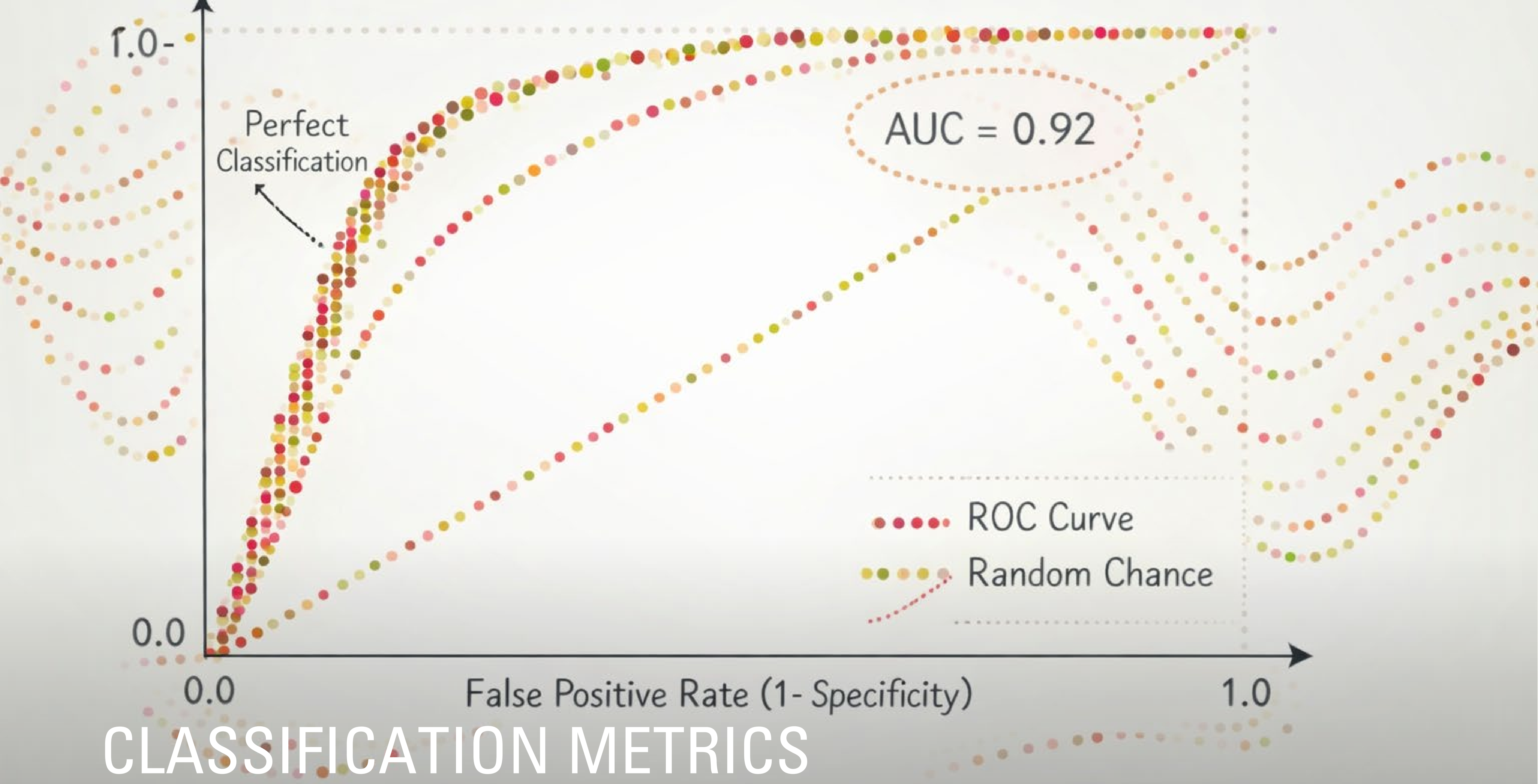


CLASSIFICATION METRICS

REGRESSION VS CLASSIFICATION (REVIEW)

- Regression and Classification are both supervised learning methods
- Regression
  - The target or response variable (y) is numeric
  - Model predicts the *value* of y
  - Prediction accuracy is measured by how close the predicted value is to the truth, such as

$$(y_i - \hat{y}_i)^2 \text{ or } |y_i - \hat{y}_i|^2$$

REGRESSION VS CLASSIFICATION (REVIEW)

- Regression and Classification are both supervised learning methods
- Classification
  - The target or response variable (y) is categorical
  - Model predicts category of y (or in some cases the probability of being in a particular class, given the features)
  - Prediction accuracy can be measured by the error rate,

$$\mathbf{ErrorRate} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{y_i \neq \hat{y}_i\}$$

ACCURACY

$$accuracy = \frac{Number\ of\ correct\ descisions\ made}{Total\ number\ of\ decisions\ made}$$

- Accuracy assumes equal costs for all types of errors
- When is accuracy not a good fitness metric?
  - When there is class skew (e.g., imbalance)
  - In these cases good accuracy could mean always predicting the majority class!

CONFUSION MATRIX

		Predicted Output	
		Positive (PP)	Negative (PN)
True Values	Positive (P)	True Positives TP	False Negatives FN
	Negative (N)	False Positives FP	True Negatives TN

PRECISION

		Predicted Output	
		Positive (PP)	Negative (PN)
True Values	Positive (P)	True Positives TP	False Negatives FN
	Negative (N)	False Positives FP	True Negatives TN

- Of those that are **predicted to be positive**, the percent that are actually positive
- Describes how good the model is at predicting the positive class
- Also call **positive predictive value**

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

Under what conditions is high **precision** the most appropriate performance objective for a classification model?

RECALL

		Predicted Output	
		Positive (PP)	Negative (PN)
True Values	Positive (P)	True Positives TP	False Negatives FN
	Negative (N)	False Positives FP	True Negatives TN

- Of those that are **actually positive**, the percent that are predicted to be positive
- The proportion of positives that are correctly identified
- Also called **sensitivity** or **true positive rate**

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

Under what conditions is high **recall** the most appropriate performance objective for a classification model?

F1 SCORE

- The precision & recall are often combined into one metric called F_1
- F_1 is the **harmonic mean** of the precision and recall

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2\left(\frac{precision \cdot recall}{precision + recall}\right)$$

- The F_1 score favors classifiers with similar precision and recall scores

F_β SCORE

- The F_1 score can be generalized to give more weight to either the precision or recall

$$F_\beta = (1+\beta^2)\left(\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}\right)$$

$0 < \beta < 1 \Longrightarrow$ More weight given to the precision

$\beta > 1 \Longrightarrow$ More weight given to the recall

$\beta = 1 \Longrightarrow$ Equal weight given to precision and recall (F_1)

OTHER MEASURES

Recall / Sensitivity / True Positive Rate

	Predicted Output: Positive (PP)	Negative (PN)
True Values: Positive (P)	TP	FN
Negative (N)	FP	TN

$$\frac{TP}{TP + FN}$$

F_1

Precision / Positive Predictive Rate

	Predicted Output: Positive (PP)	Negative (PN)
True Values: Positive (P)	TP	FN
Negative (N)	FP	TN

$$\frac{TP}{TP + FP}$$

ROC Curve & AUC

Negative Predictive Value

	Predicted Output: Positive (PP)	Negative (PN)
True Values: Positive (P)	TP	FN
Negative (N)	FP	TN

$$\frac{TN}{TN + FN}$$

Specificity / True Negative Rate

	Predicted Output: Positive (PP)	Negative (PN)
True Values: Positive (P)	TP	FN
Negative (N)	FP	TN

$$\frac{TN}{TN + FP}$$

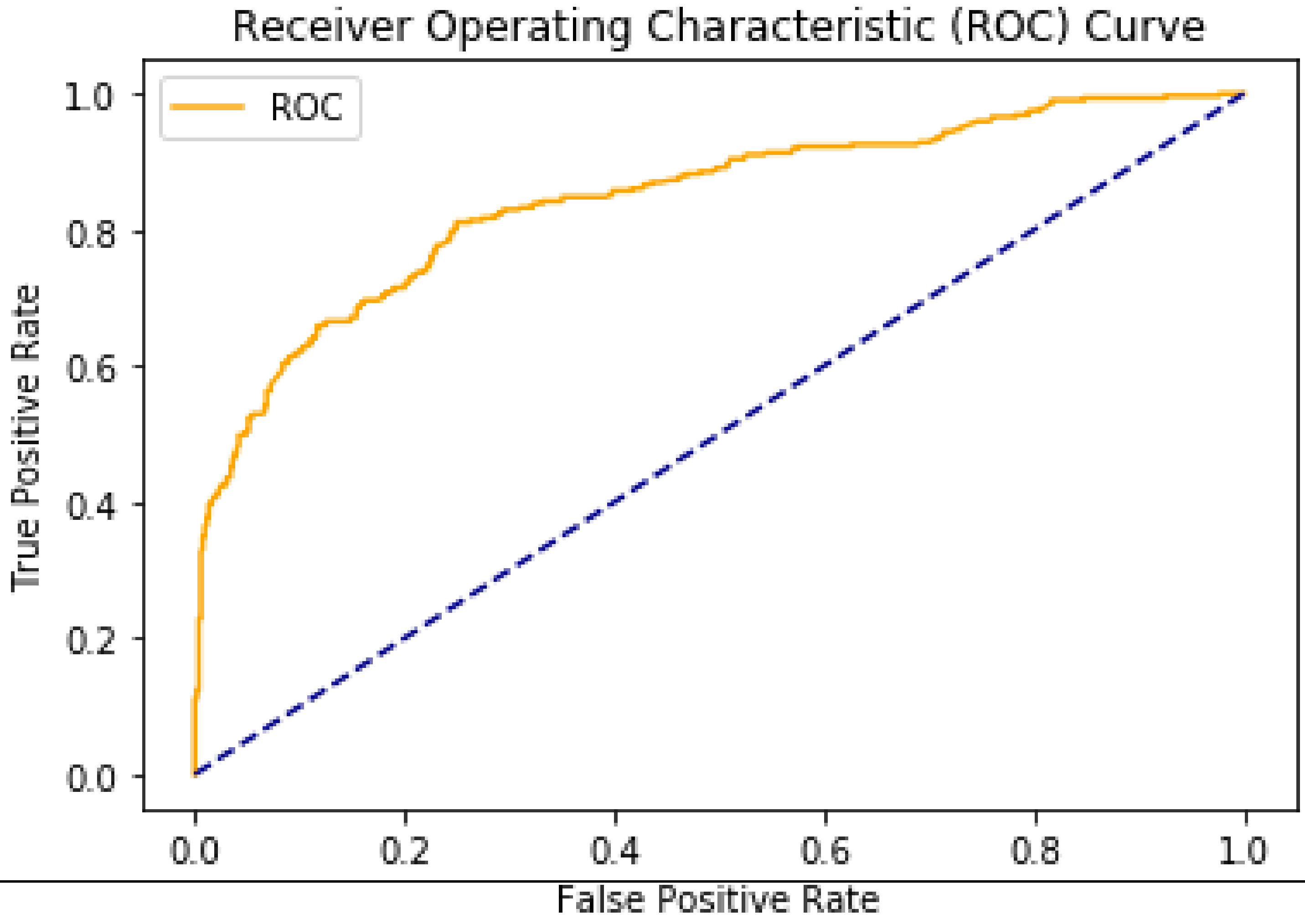
ROC CURVE AND AUC

- ROC stands for Receiver Operating Curve
  - Developed in WWII to model false positive and false negative detections of radar operators
  - It plots the True Positive Rate (sensitivity) versus the False Positive Rate (1 - specificity) for different classification thresholds
- AUC stands for Area Under the (ROC) Curve
  - It is a single measure for summarizing the ROC curve
    - AUC =1 $\Longrightarrow$ perfect model
    - AUC = 0.5 $\Longrightarrow$ random model

BUILDING THE ROC CURVE

- Many classifier actually predict a probability or score rather than a classification
- Then the classification is made by setting a threshold
  - For example: if $P(y_i = positive) > p$ then classify y_i as positive (for $0 < p < 1$)
- The ROC curve looks at the true positive rate versus the false positive rate for a range of thresholds

EXAMPLE ROC CURVE



METRICS IN SCIKIT-LEARN

FUNCTION NAMES

https://scikit-learn.org/stable/api/sklearn.metrics.html#

- `accuracy_score(y_true, y_pred)`
  (default for many functions)
- `classification_report(y_true, y_pred)`
- `confusion_matrix(y_true, y_pred)`
- `f1_score(y_true, y_pred)`
- `precision_score(y_true, y_pred)`
- `recall_score(y_true, y_pred)`
- `roc_auc_score(y_true,` **`y_score)`** **`Note:(`**`The` second argument must be a score (e.g. probability) and not a class prediction

.PREDICT() VS .PREDICT_PROBA()

- For classification estimators, there are two ways to product predictions
- `.predict()` will produce an 1D array with n rows with the class prediction
    - Class with the highest probability
- `.predict_proba()` will produce a 2D array with n rows and n-class columns
    - Each column is the probability of the row observation being in class 0 to n-class

CLASSIFICATION REPORT

- This special function will show precision, recall, f1, and accuracy for all class (treat each class as positive class)

```
print(classification_report(y_test, yhat))
```

✓ 0.0s

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       980
           1       0.91      1.00      0.95      1135
           2       0.98      0.91      0.95      1032
           3       0.96      0.96      0.96      1010
           4       0.97      0.94      0.95       982
           5       0.96      0.95      0.96       892
           6       0.96      0.98      0.97       958
           7       0.95      0.94      0.94      1028
           8       0.99      0.91      0.95       974
           9       0.93      0.95      0.94      1009

    accuracy                           0.95     10000
   macro avg       0.96      0.95      0.95     10000
weighted avg       0.95      0.95      0.95     10000
```

MULTI-CLASS CLASSIFICATION

MULTI-CLASS STRATEGIES

- Some classifiers (e.g., Naive Bayes, KNN) are capable of handling multiple classes directly
- Other classifiers (e.g., binary logistic regression) are binary classifiers
- For binary classifiers, there are two strategies for extending to multiple classes:
    - One-versus-all (OvA) (sometimes called One-versus-rest (OvR))
    - One-versus-one (OvO)

ONE VERSUS ALL (OVA)

- For each class, train a model that classify that class versus all the rest:
    - Classify **1** versus **not 1**
    - Classify **2** versus **not 2**
    - etc.
- Classify a value into the classier whose classifier outputs the highest score (e.g., predictive probabilities)

ONE VERSUS ONE (OVO)

- Train a binary classifier for every pair of classes
  - **1** vs **2**
  - **1** vs **3**
  - **2** vs **3**
  - **4** vs **5**
  - etc
- Classify value into the class that wins the most duels

OVA VERSUS OVO

- Suppose there are q classes
- OvO requires training q(q-1)/2 classifiers
  - For the MNIST example this means 10(10-1)/2 = 45 classifiers
  - However, each classifier needs only use a small subset of the data
- In general OvA is the preferred strategy (only trains q classifiers)
- But for classifiers that scale poorly with the size of the training set (e.g., SVM) OvO is preferred

SOFTMAX FUNCTION

- A mathematical function that converts a vector of real-valued scores into a probability distribution
- Used to convert real-valued outputs to "probabilities"
- Can be used as a way to extend logistic regression
- Often used in multi-class NN

$$p(c_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

z_i is a raw score for the class (for example the class logit)

Probability of class c_i

Sum over all classes