Model Evaluation

Chantri Polprasert

CPDSAI

Objectives

- Understand ML model evaluation for classification and regression problems
- Able to explain some characteristics of each evaluation model
- Appropriately apply evaluation model to different problems
- Apply evaluation model concept using Sklearn library

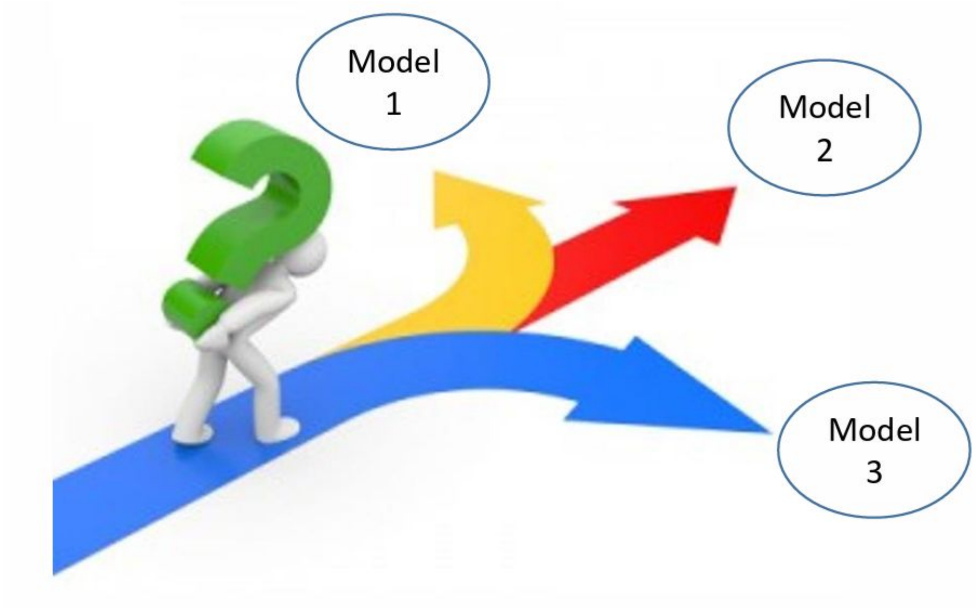
Topics to cover

- Evaluation metrics for classification
    - Accuracy
    - Confusion matrix and its primary and secondary metrics
    - Precision and Recall relationship
- Evaluation metrics for regression
    - Python commands

Evaluation metrics for classifier

- Accuracy
- Confusion matrix
- Classification error
- Precision-recall
- Specificity
- F-1 score

- Let's first consider a binary response in which two types of errors can be made.

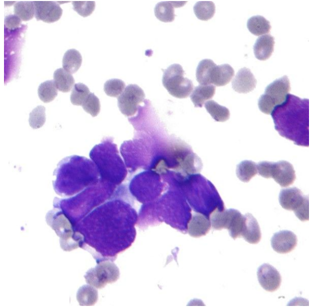


Accuracy

Binary classification revisit

- There are only two possible classes or outcomes to predict (e.g. yes or no)
- Class 1: positive class, class of interest (depend on our definition)
- Class 0: negative class, not interested
- Some examples of Binary Classification are:
    - Predicting whether a customer will default his mortgage repayments,
    - Predicting whether a person will exit his internet plan upon expiry of his contract
    - Predicting whether a credit-card transaction is normal or the result of card theft

Example



- Malignant classification (binary classification)
  - Breast cancer dataset
  - From 269 samples, 203 of them are malignant and 66 are benign tumors
  - Assume that our binary classification model performance is presented in the following confusion matrix:
  - What's the accuracy of this classification model?

		Predicted		
		Malignant	Benign	Total
Actual	Malignant	201	2	203
	Benign	5	61	66
	Total	63	206	269

Confusion matrix: a specific table layout that allows visualization of the performance of an algorithm

https://archive.ics.uci.edu/ml/machine-learning%20databases%20/breast%20cancer-wisconsin/wdbc.data

Classification accuracy vs misclassification rate

	Predicted Yes	Predicted No
Actual Yes	a = True Positive	b = False Negative
Actual No	c = False Positive	d = True Negative

$$\text{Classification Accuracy} = \frac{a+d}{a+b+c+d} = \frac{a+d}{n}$$

$$\text{Misclassification Rate} = \frac{b+c}{a+b+c+d} = \frac{b+c}{n}$$

Is accuracy enough to evaluate classification performance?

- What's the **recall** of the fortune teller (classification model) which predicts that Halley's comet will not be visible every year for 80 years?

Halley, officially designated 1P/Halley, is a short-period comet visible from Earth every 75–79 years.

https://en.wikipedia.org/wiki/Halley%27s_Comet

Unbalanced Example

Is a model that provides 0.9 classification accuracy good?

	Predicted	
	Yes	No
Actual Yes	87	1
Actual No	10	2

Classification accuracy = 89/100.

- Only 2 of the 12 that were classified to the 'no' group were correct.
- The 'no error rate' is 10/12 or 83%.

Problems with accuracy :Example1

- For unbalanced (imbalanced) classes, high classification accuracy (equivalently, low misclassification rates) can be deceiving.
- While these measures are easily extended to multi-class scenarios, the problem with unbalanced classes remains.
- Is 98% classification accuracy better that those with 0.82?

Classifier 1

	Predicted	
	Yes	No
Actual Yes	87	1
Actual No	1	11

Classifier 2

	Predicted	
	Yes	No
Actual Yes	79	9
Actual No	9	3

Problems with accuracy: Example2

- Is 98% classification accuracy better than those with 0.82?

Classifier 1

	Predicted Yes	Predicted No
Actual Yes	88	0
Actual No	2	10

Classifier 2

	Predicted Yes	Predicted No
Actual Yes	70	18
Actual No	1	12

Example 2

- If it’s very important to correctly classify NOs we may prefer Classifier 2 over Classifier 1 (e.g. trying to identify email as spam)
- If it’s very important to correctly classify YESs we may prefer Classifier 1 over Classifier 2 (eg identifying people who has a deadly disease that requires treatment)
- What about regression? What’s wrong if we focus on minimizing the MSE (RSS)?

Issues with accuracy

- We may consider weighted metrics wherein the classes are weighted differently in order to emphasize more "cost" to misclassification of one group over the other.
- Another common approach is to consider metrics that may provide more insight than simply accuracy and misclassification rates alone.

LogisticRegression

```
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False,
tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None,
random_state=None, solver='lbfgs', max_iter=100, multi_class='deprecated', verbose=0,
warm_start=False, n_jobs=None, l1_ratio=None)
```

[source]

Sklearn uses .score()

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    cancer.data, cancer.target, stratify=cancer.target, random_state=42)
logreg = LogisticRegression(solver = 'liblinear').fit(X_train, y_train)
print(f"Training set score: {logreg.score(X_train, y_train):.3f}")
print(f"Test set score: {logreg.score(X_test, y_test):.3f}")
```

```
Training set score: 0.953
Test set score: 0.958
```

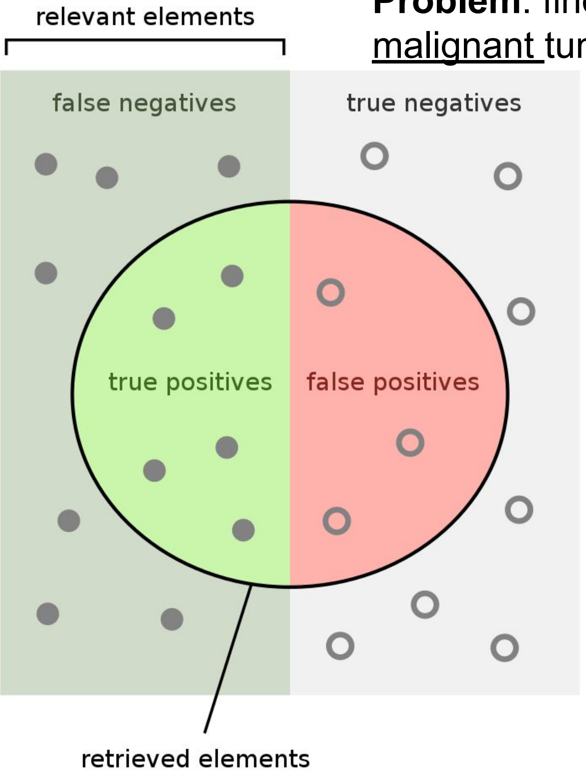
Primary Measures

Problem: find benign or malignant tumors

Confusion matrix

	Predicted negative	Predicted positive
True negative	TN	FP
True positive	FN	TP

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$



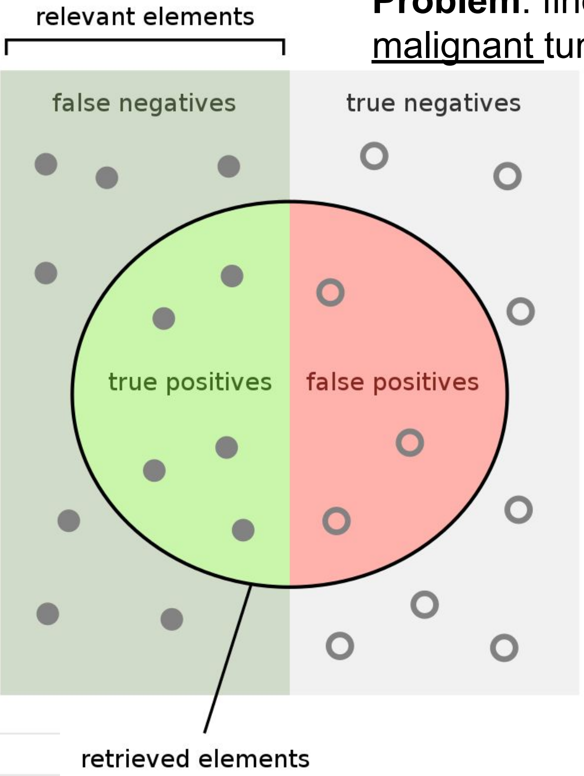
https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

- TP: True Positive
- FP: False Positive (Type I error)
- TN: True Negative
- FN: False Negative (Type II error)

Problem: find benign or <u>malignant </u>tumors

Confusion matrix

	Predicted negative	Predicted positive
True negative	TN	FP
True positive	FN	TP

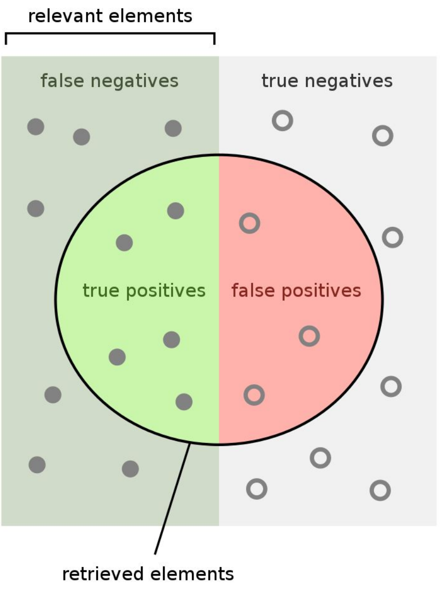


https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

		Predicted		
		Malignant	Benign	Total
Actual	Malignant	201 (TP)	2 (FN)	203
	Benign	5 (FP)	61 (TN)	66
	Total	206	63	269

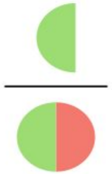
Precision (Positive Predicted Value (PPV))

- How many retrieved items are relevant?
- A ratio of predicted "Yes" that are actually "Yes"
- Higher precision suggests that the model is good at avoiding false positives. It focuses on making accurate positive predictions (i.e. the cost of false positives are relatively low or manageable.)



		Predicted	
		Malignant	Benign
Actual	Malignant	201 (TP)	2 (FN)
	Benign	5 (FP)	61 (TN)

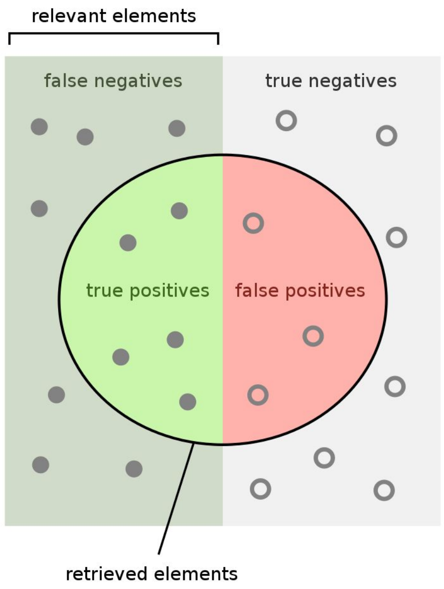
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} =$$



What's the precision?

Recall (Sensitivity, coverage, True positive rate (TPR))

- How many relevant items are retrieved?
- The proportion of actual “Yes”s that were predicted “Yes”.
- Only interested in positive population
- Higher recall suggests that the model is good at avoiding false negatives.



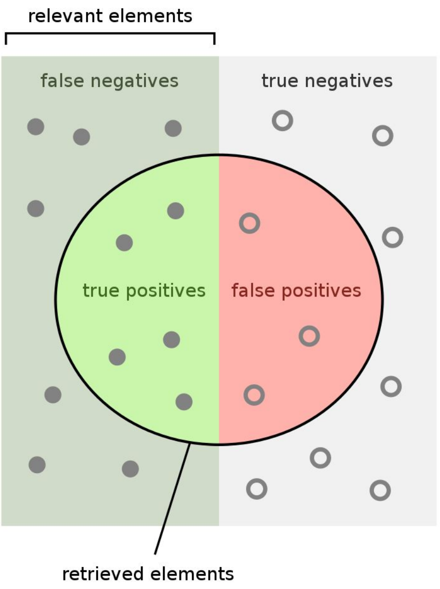
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} =$$

		Predicted	
		Malignant	Benign
Actual	Malignant	201 (TP)	2 (FN)
	Benign	5 (FP)	61 (TN)

- What’s the recall of this classification model?
- What kind of problems that we are concern with recall?

Specificity, True Negative Rate (TNR),

- The proportion of actual "No"s that were predicted "No"
- Specificity focuses on avoiding false positives, and is concerned with correctly identifying all negative instances.



		Predicted	
		Malignant	Benign
Actual	Malignant	201 (TP)	2 (FN)
	Benign	5 (FP)	61 (TN)

$$\texttt{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

Is recall enough to evaluate classification performance?

- What's the **recall** of the fortune teller (classification model) which predicts that Halley's comet will not be visible every year for 80 years?

Halley, officially designated 1P/Halley, is a short-period comet visible from Earth every 75–79 years.

https://en.wikipedia.org/wiki/Halley%27s_Comet

Is recall enough to evaluate classification performance?

- What's the **recall and precision** of the fortune teller (classification model) which predicts that Halley's comet will be visible every year for 80 years?

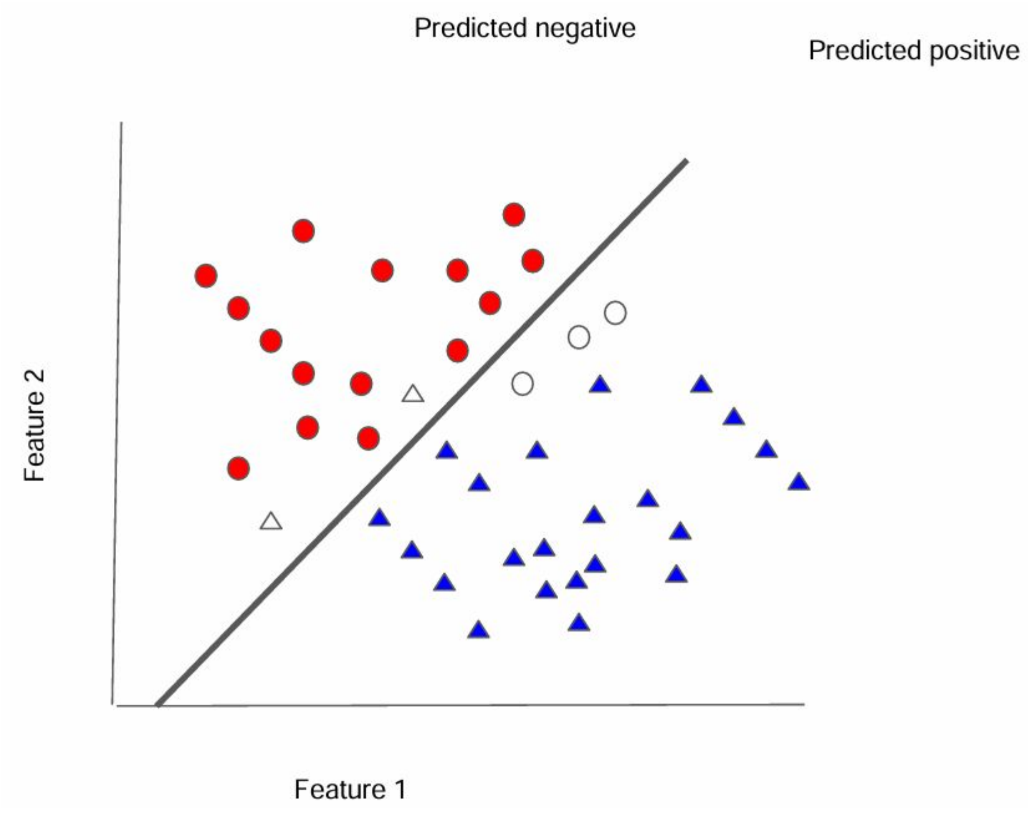
Halley, officially designated 1P/Halley, is a short-period comet visible from Earth every 75–79 years.

https://en.wikipedia.org/wiki/Halley%27s_Comet

Exercise

- Assume that there are 50 stones and 4 diamonds in the urn. Your job is to find the classifier that can accurately find the diamonds. What’s the precision and recall of the classifier when
  - The classifier always predicts the diamond.
  - The classifier is very selective in predicting the diamond.
  - Less strict?

Relationship between precision and recall



FN = 2; TP=22; FP=3;

$$Precision = \frac{TP}{TP + FP} = \frac{22}{22 + 3} = 0.88$$

$$Recall = \frac{TP}{TP + FN} = \frac{22}{22 + 2} = 0.92$$

▲ True positive

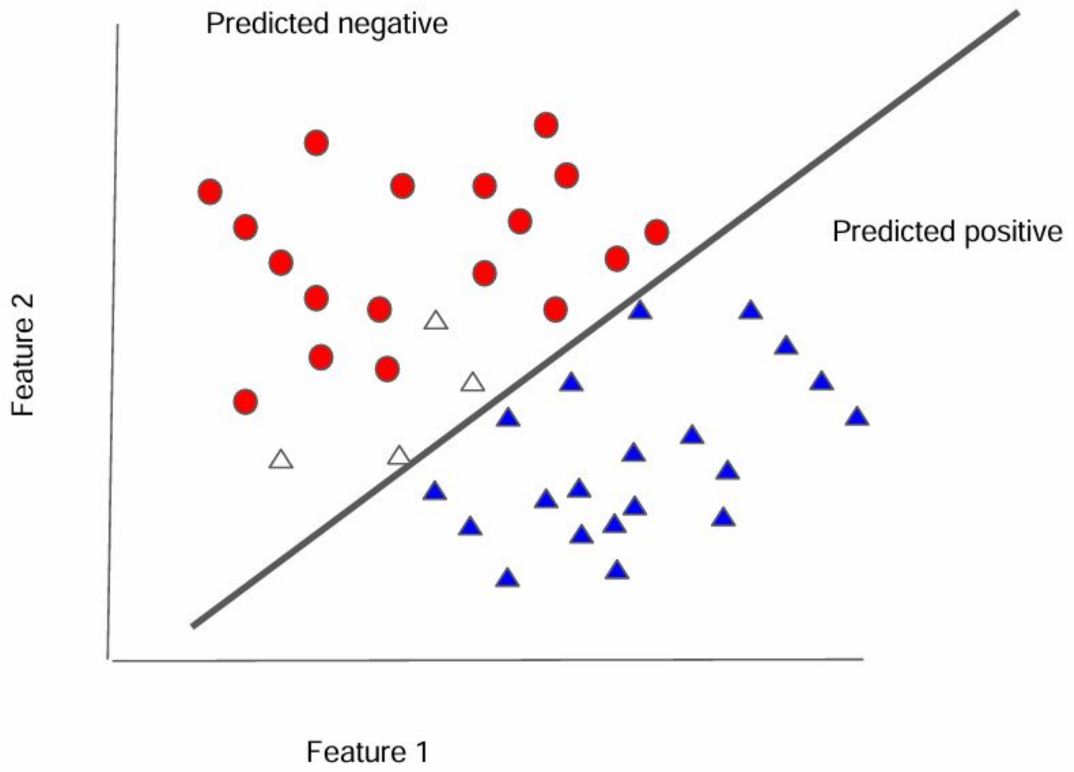
● True negative

○ False positive

△ False negative

High precision and low recall

Increase threshold to improve precision



FN = 4; TP=20; FP=0;

$$Precision = \frac{TP}{TP + FP} = \frac{20}{20 + 0} = 1$$

$$Recall = \frac{TP}{TP + FN} = \frac{20}{20 + 4} = 0.83$$

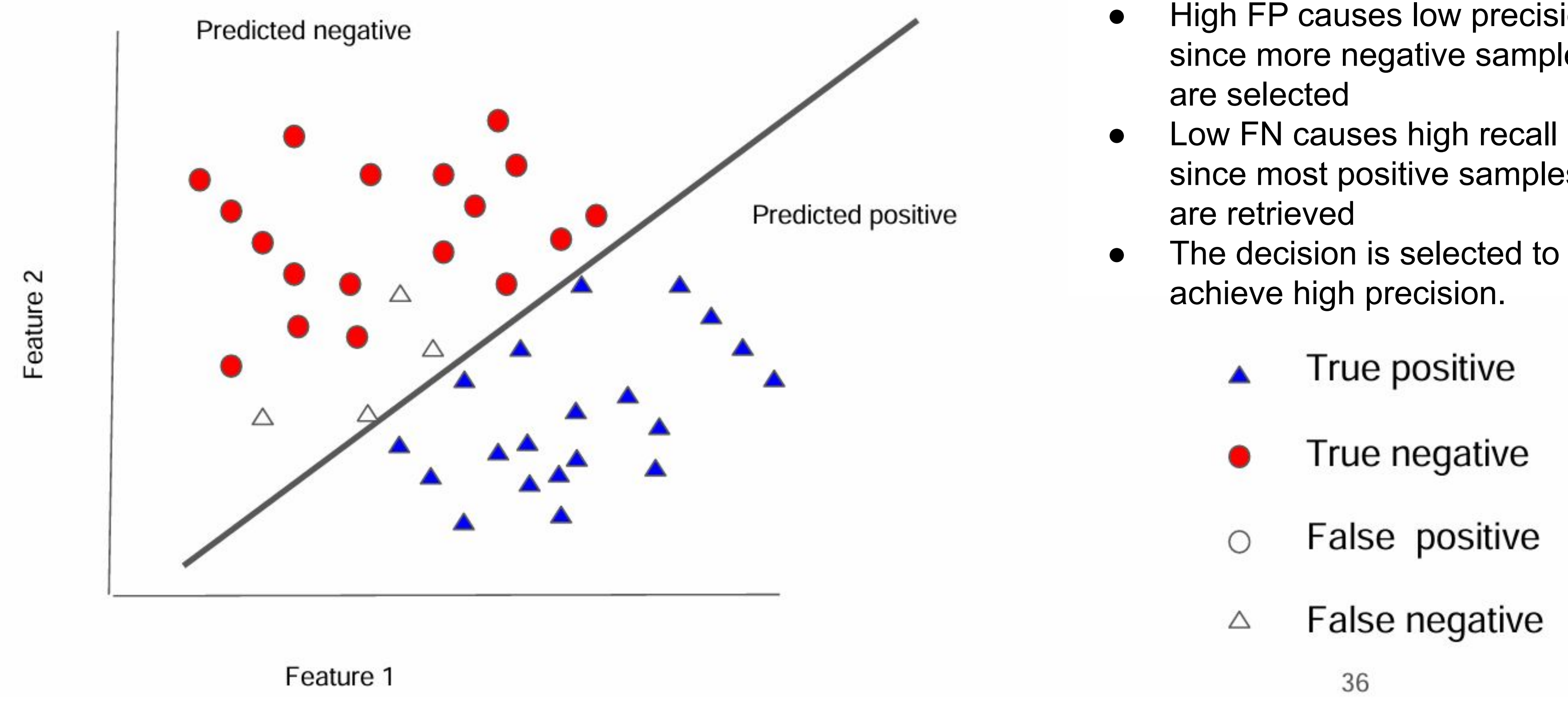
▲ True positive

● True negative

○ False positive

△ False negative

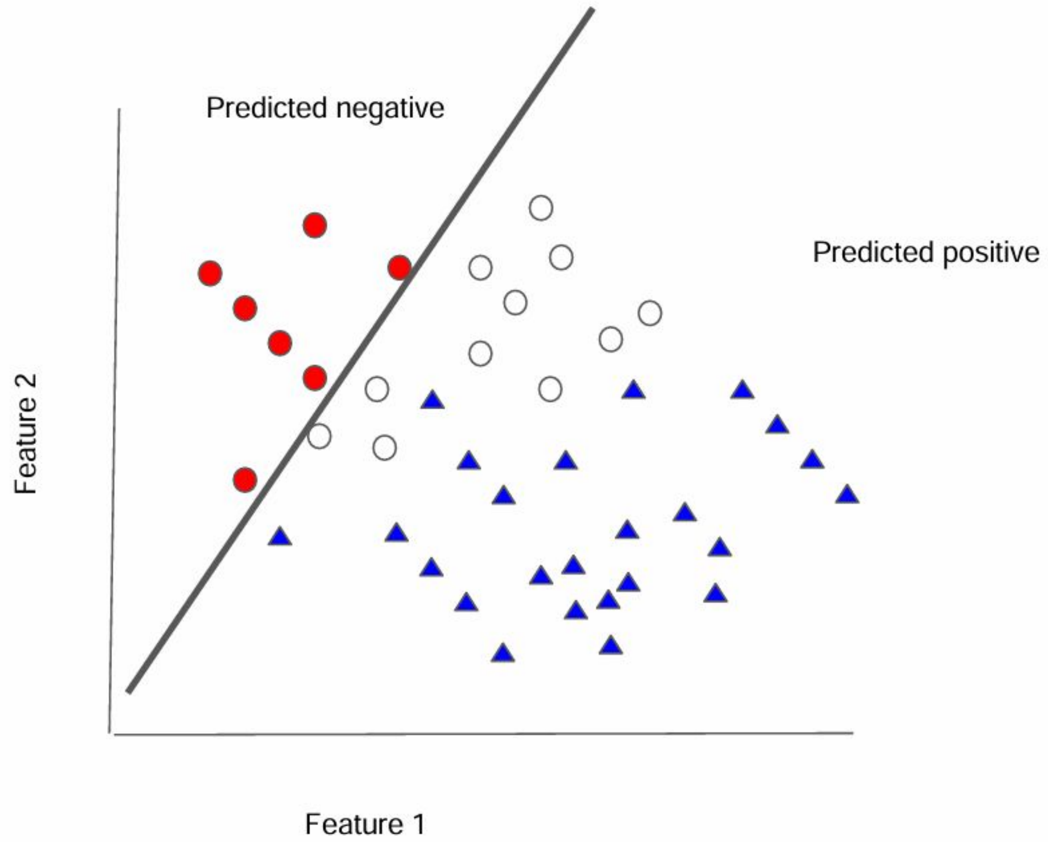
High precision and low recall



- High FP causes low precision since more negative samples are selected
- Low FN causes high recall since most positive samples are retrieved
- The decision is selected to achieve high precision.

Low precision and high recall

Decrease threshold to improve recall



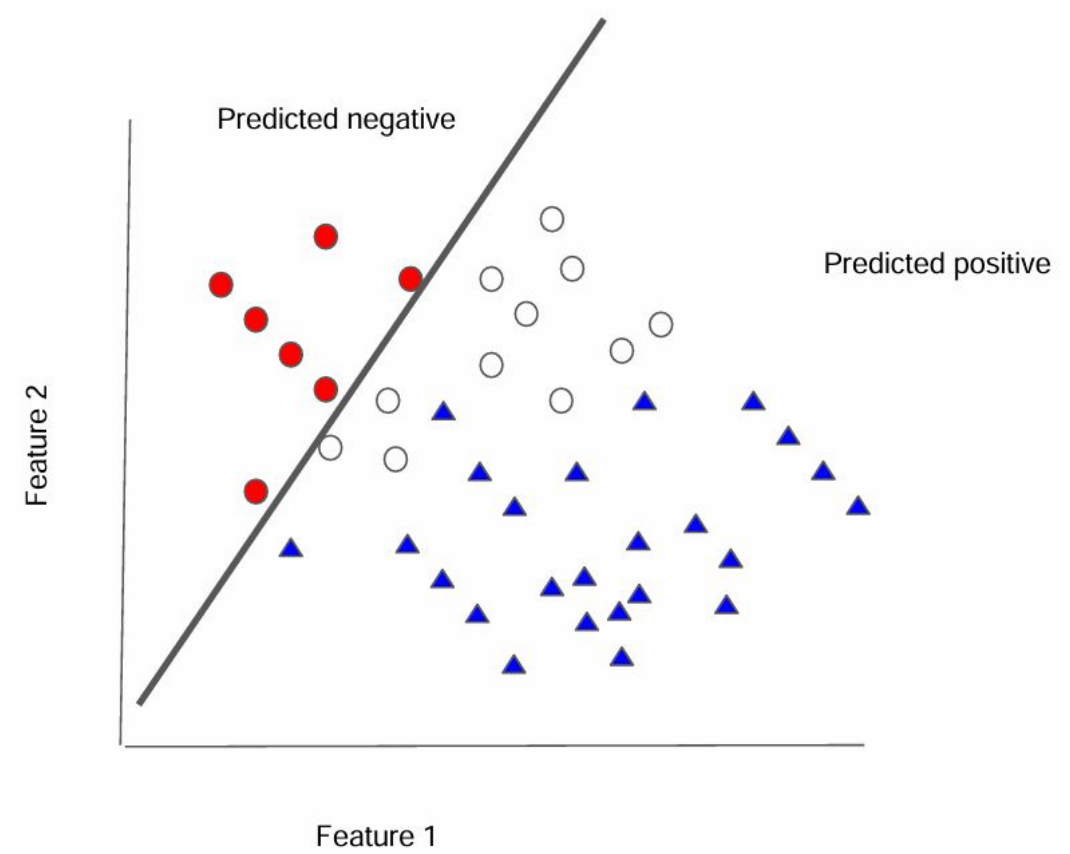
FN = 0; TP=24; FP=11;

$$Precision = \frac{TP}{TP+FP} = \frac{24}{24+11} = 0.68$$

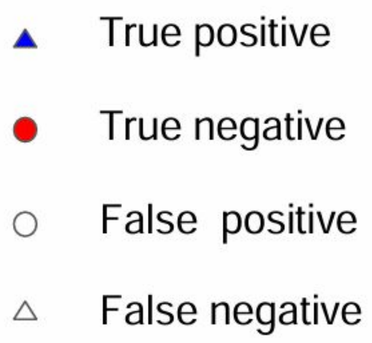
$$Recall = \frac{TP}{TP+FN} = \frac{24}{24+0} = 1$$

- ▲ True positive
- ● True negative
- ○ False positive
- △ False negative

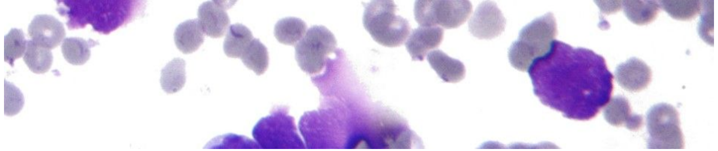
Low precision and high recall



- High FP causes low precision since more negative samples are selected
- Low FN causes high recall since most positive samples are retrieved
- The boundary is selected to achieve high recall



Low threshold to improve recall



Breast Cancer Wisconsin (Diagnostic) Data Set

```
lr_predicted = lr.predict(X_test)
print('Logistic regression (threshold=50%)\n',
      classification_report(y_test, lr_predicted, target_names = ['not 1', '1']))

y_predicted = y_proba_lr[:, 1] > .15

print('Logistic regression (threshold=15%)\n',
      classification_report(y_test, y_predicted, target_names = ['not 1', '1']))
```

```
Logistic regression (threshold=50%)
              precision    recall

      not 1       0.91      0.98
          1       0.99      0.94
```

```
Logistic regression (threshold=15%)
              precision    recall

      not 1       0.98      0.89
          1       0.94      0.99
```

High threshold to improve precision

```
lr_predicted = lr.predict(X_test)
print('Logistic regression (threshold=50%)\n',
      classification_report(y_test, lr_predicted, target_names = ['not 1', '1']))

y_predicted = y_proba_lr[:, 1] > .85

print('Logistic regression (threshold=15%)\n',
      classification_report(y_test, y_predicted, target_names = ['not 1', '1']))
```

```
Logistic regression (threshold=50%)
              precision    recall

       not 1       0.91      0.98
           1       0.99      0.94
```

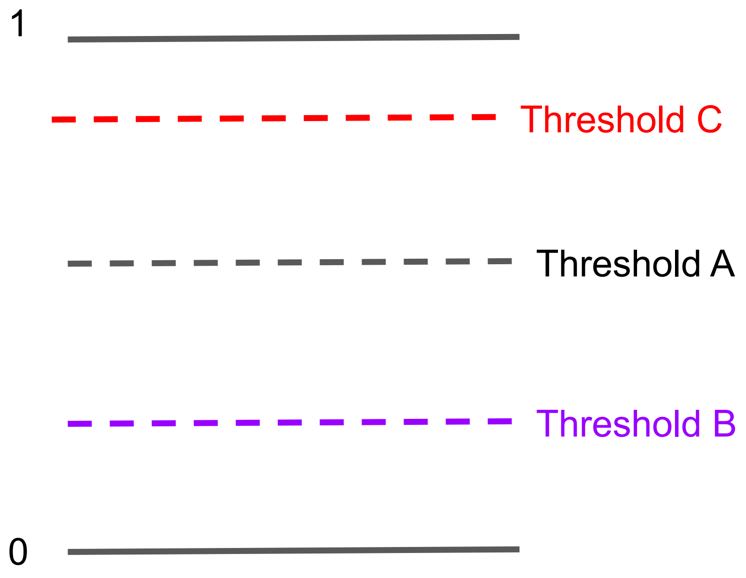
```
Logistic regression (threshold=85%)
              precision    recall

       not 1       0.79      1.00
           1       1.00      0.84
```

Precision vs Recall tradeoff

- Jobs that require high recall: (To retrieve many relevant samples while allowing some non-relevant samples)
  - Search and information extraction in legal discovery
  - Malignant tumor detection
  - Used with experts to filter out FP
- Jobs that require high precision: (To obtain relevant samples while minimize non-relevant samples)
  - Search engine
  - Spam detection

Exercise



Given (P0,R0) and (P1, R1), the precision and recall of class 0 and class 1 at threshold A.

- What's the relationship between (P0,R0) and (P1, R1) at threshold B compared to (P0,R0) and (P1, R1) at threshold A?
- What's the relationship between (P0,R0) and (P1, R1) at threshold C compared to (P0,R0) and (P1, R1) at threshold A?

		Predicted condition		Sources: [4][5][6][7][8][9][10][11][12] view · talk · edit	
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP),** **hit**	**False negative (FN),** **type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP),** **type I error, false alarm, overestimation**	**True negative (TN),** **correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN}$ $= 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

The zoo

https://en.wikipedia.org/wiki/Precision_and_recall

Secondary Measures

F1-score

- Harmonic means of precision and recall
- Balance between precision and recall

$$F_1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \times Recall}{(\beta^2 \cdot Precision) + Recall} = \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta \cdot FN + FP}$$

The Fbeta-measure is a generalization of the F-measure that adds a configuration parameter called beta

- Large beta (> 1) gives more weight to recall
- Small beta (< 1) gives more weight to precision

F1-score vs Precision-Recall

```
lr_predicted = lr.predict(X_test)
print('Logistic regression (threshold=50%)\n',
      classification_report(y_test, lr_predicted, target_names = ['not 1', '1']))

y_predicted = y_proba_lr[:, 1] > .85

print('Logistic regression (threshold=85%)\n',
      classification_report(y_test, y_predicted, target_names = ['not 1', '1']))
```

```
Logistic regression (threshold=50%)
              precision    recall  f1-score   support

      not 1       0.91      0.98      0.95        53
          1       0.99      0.94      0.97        90
```

F1-score tends to be in the middle between P &R

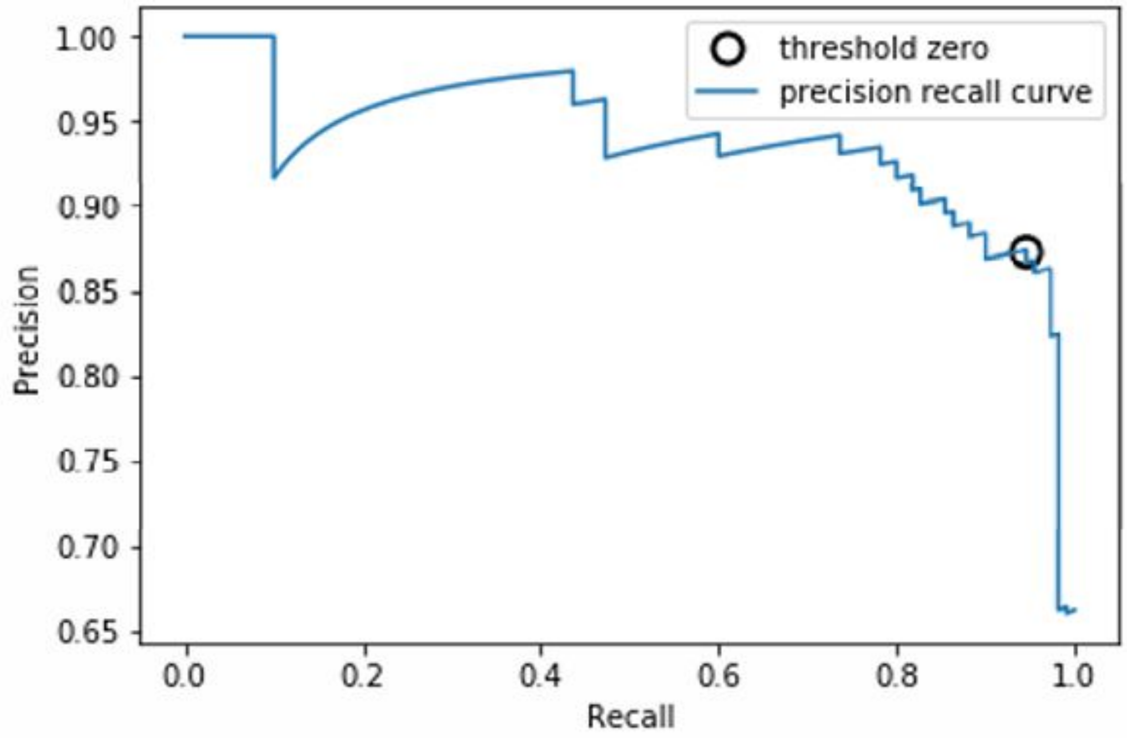
Precision-Recall (PR) curve

- X-axis: Recall
- Y-axis: Precision

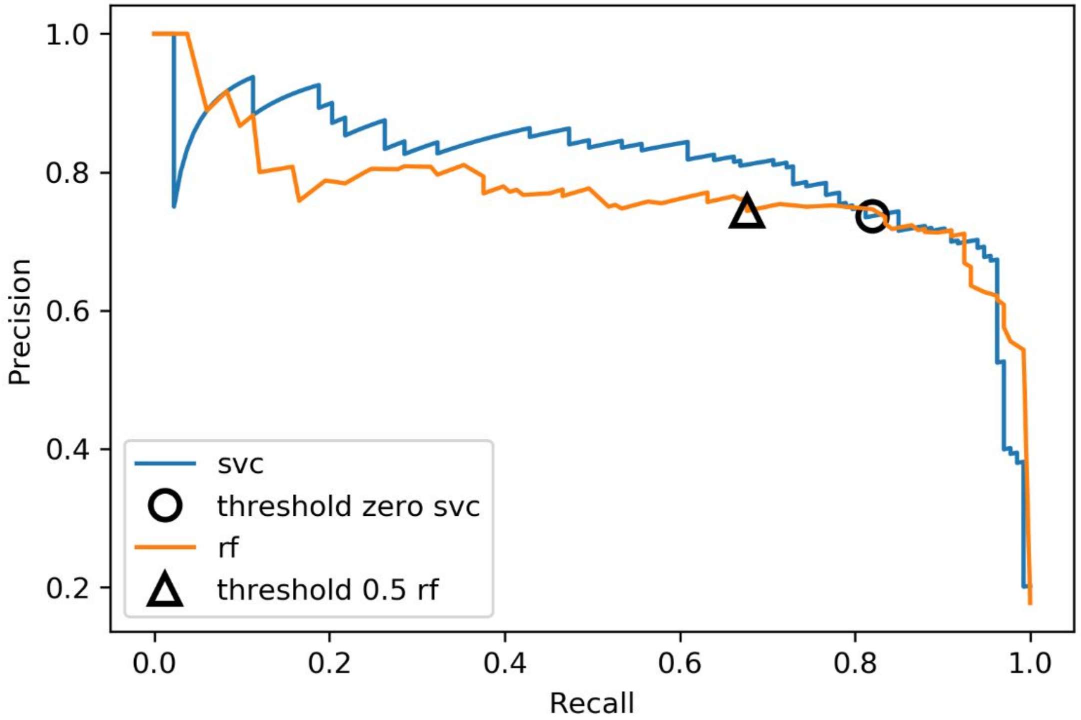
Top right corner:

- The “ideal” point
- Precision = 1.0
- Recall = 1.0

- F1 score is considered at one threshold.
- PR curve considers for every threshold value.



Comparing Random Forest (RF) and Support Vector Classifier (SVC)



Which one is better?
Say you want a model with 70% recall.

Average Precision (AP)

- AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.
- Related to area under the precision-recall curve (with step interpolation)

Precision at threshold k

$$\text{AveP} = \sum_{k=1}^{n} P(k)\Delta r(k)$$

Change in recall between k and k-1

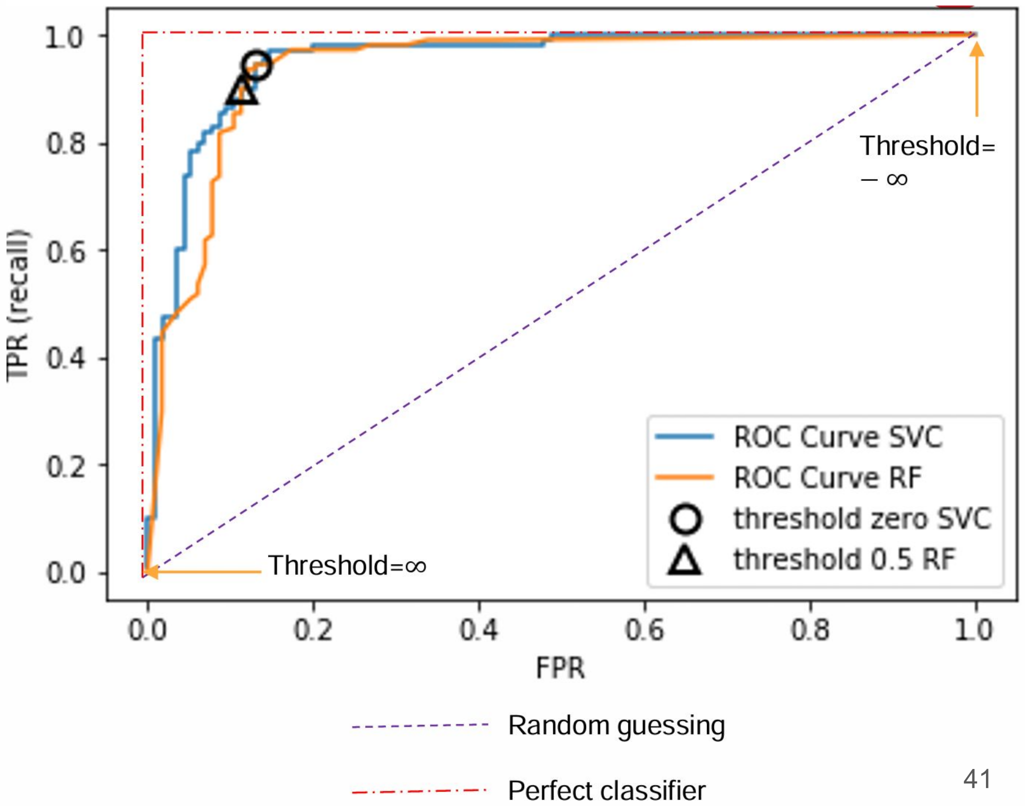
Sum over data points, ranked by decision function

4

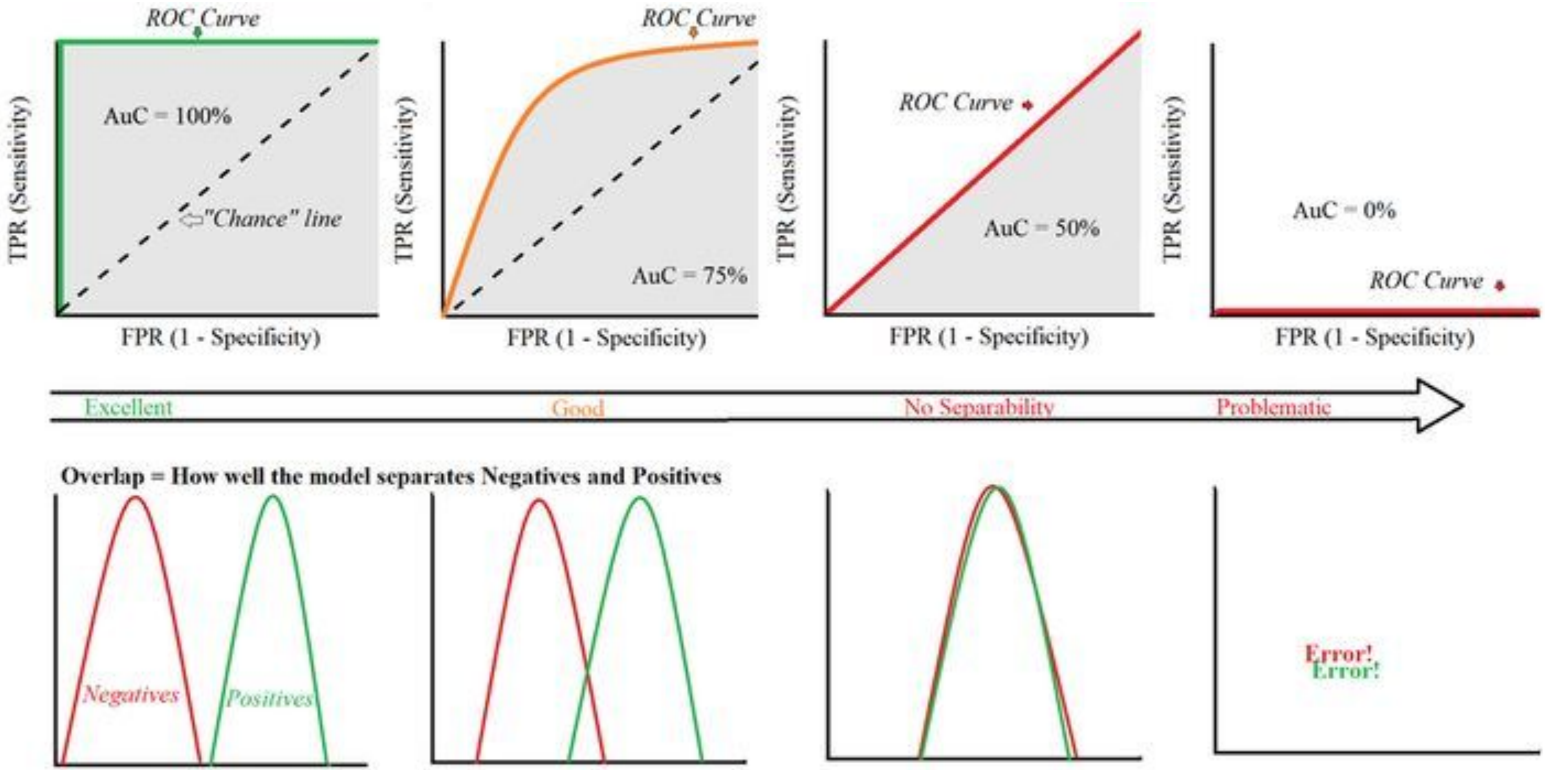
ROC Curves (Receiver Operating Characteristic Curve)

- X-axis: False Positive Rate
- Y-axis: True Positive Rate (Recall)

- Top left corner:
  - The “ideal” point
  - False positive rate of zero
  - True positive rate of one
- “Steepness” of ROC curves is important:
  - Maximize the true positive rate
  - while minimizing the false positive rate

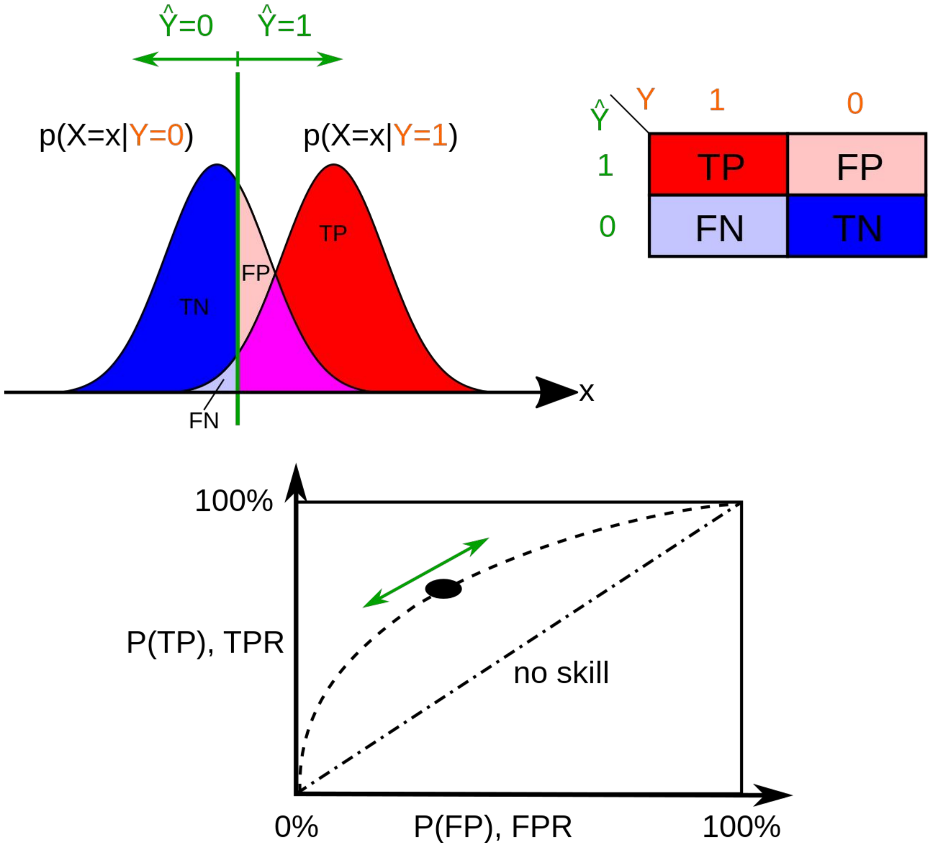


ROC Curves (Receiver Operating Characteristic Curve)



https://www.datasciencecentral.com/roc-curve-explained-in-one-picture/

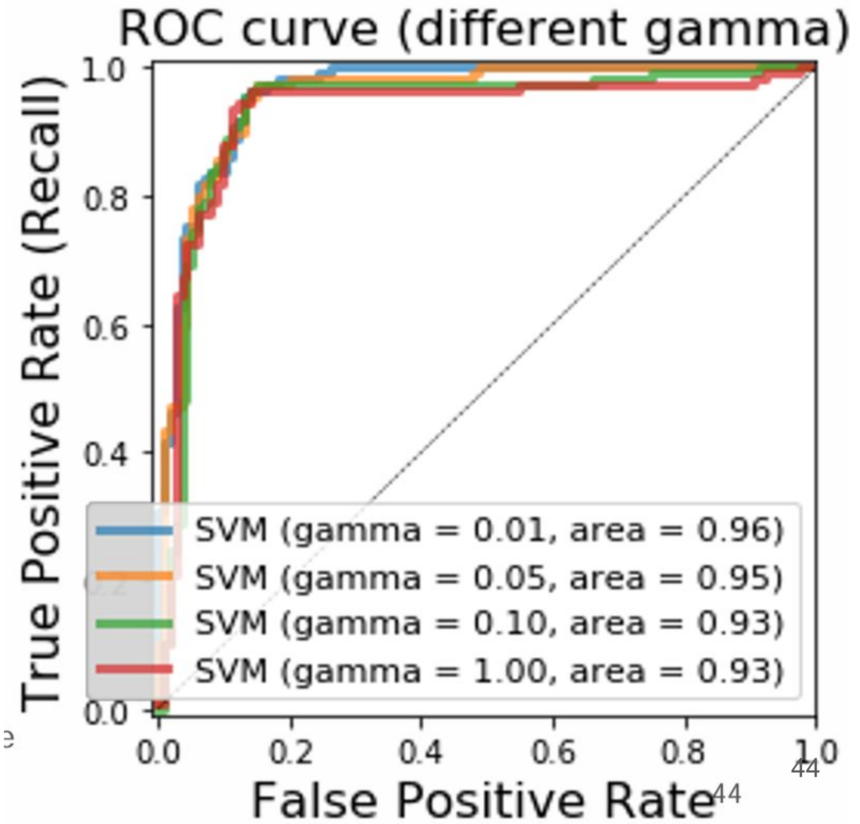
ROC Curves (Receiver Operating Characteristic Curve)



https://en.wikipedia.org/wiki/File:ROC
curves.svg

Summarizing an ROC curve in one number: Area Under the Curve (AUC)

- AUC = 0 (worst) AUC = 1 (best) AUC = 0.5 (random)
- AUC can be interpreted as:
  - 1. The total area under the ROC curve.
  - 2. The probability that the classifier will assign a higher score to a randomly chosen positive example than to a randomly chosen negative example.
- Advantages:
  - Gives a single number for easy comparison.
  - Does not require specifying a decision threshold.
- Drawbacks:
  - As with other single-number metrics, AUC loses information, e.g. about tradeoffs and the shape of the ROC curve.
  - This may be a factor to consider when e.g. wanting to compare the performance of classifiers with overlapping ROC curves



Classification model summary

- Accuracy > Ratio of correct decision
- Precision
- Recall

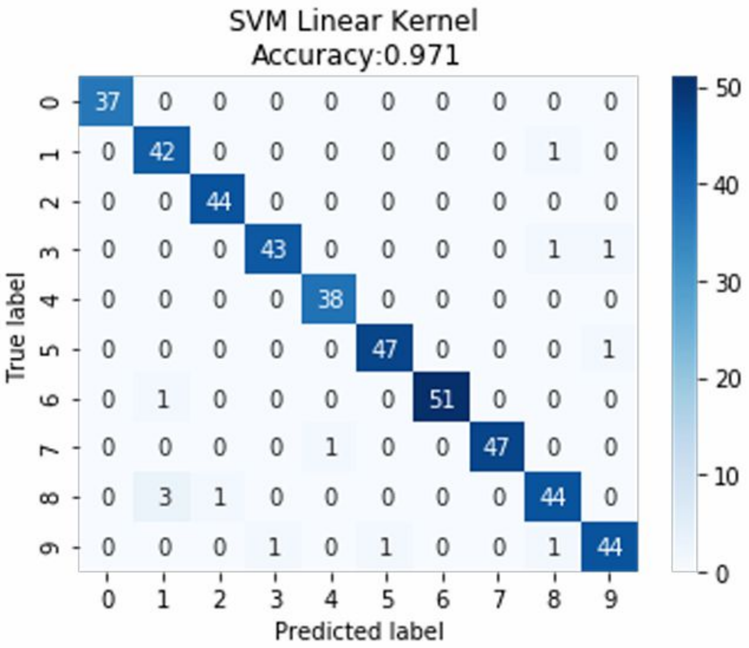
-Correct positive decision
-Tug of war with threshold

- F1-score
- Precision-Recall Curve
- ROC Curve

Vary threshold

Multiclass classification

Multi-Class Confusion Matrix




```
svm = SVC(kernel = 'linear').fit(X_train_mc, y_train_mc)
svm_predicted_mc = svm.predict(X_test_mc)
confusion_mc = confusion_matrix(y_test_mc, svm_predicted_mc)
```

Classification Report: One-vs-Rest

	precision	recall	f1-score	support
0	1.00	0.65	0.79	37
1	1.00	0.23	0.38	43
2	1.00	0.39	0.56	44
3	1.00	0.93	0.97	45
4	0.14	1.00	0.25	38
5	1.00	0.33	0.50	48
6	1.00	0.54	0.70	52
7	1.00	0.35	0.52	48
8	1.00	0.02	0.04	48
9	1.00	0.55	0.71	47
accuracy			0.49	450
macro avg	0.91	0.50	0.54	450
weighted avg	0.93	0.49	0.54	450

Macro Average

$$\text{Macro} = \frac{1}{|L|} \sum_{l \in L} R(y, \widehat{y})$$

- The arithmetic mean of the individual class related to precision, memory, and f1 score
- All classes are treated equally
  - Compute metric within each class
  - Average resulting metrics across classes

Class	Recall
orange	1/5 = 0.20
lemon	1/2 = 0.50
apple	2/2 = 1.00

Macro-average Recall
(0.20 + 0.50 + 1.00) / 3 = **0.57**

Class	Predicted Class	Correct?
orange	lemon	0
orange	lemon	0
orange	apple	0
orange	orange	1
orange	apple	0
lemon	lemon	1
lemon	apple	0
apple	apple	1
apple	apple	1

Weighted Average

$$\text{Weighted} = \frac{1}{n} \sum_{l \in L} n_l R(y, \widehat{y})$$

- Each class is weighted by the number of instances in that class
- Mean per-class metric, weighted by support.

Class	Recall
orange	1/5 = 0.20
lemon	1/2 = 0.50
apple	2/2 = 1.00

Weighted-average:
= (0.2 * 5 + 0.5 * 2 + 1.0 * 2) / 9
= 0.44

Class	Predicted Class	Correct?
orange	lemon	0
orange	lemon	0
orange	apple	0
orange	orange	1
orange	apple	0
lemon	lemon	1
lemon	apple	0
apple	apple	1
apple	apple	1

Metrics for regression model

Metrics for regression model

- Sum of Squared Error

$$\texttt{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

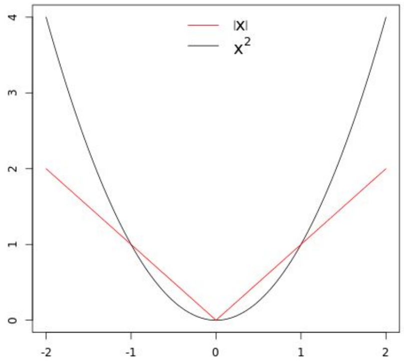
- Total Sum of Squares

$$\texttt{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- The coefficient of determination

$$R^2 = 1 - \frac{\texttt{SSE}}{\texttt{TSS}}$$

Metrics for regression model



- Mean Square Error (MSE) $$\mathtt{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
- Mean Absolute Error (MAE) $$\mathtt{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$
- Mean absolute percentage error (MAPE) $$\mathtt{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

MSE vs MAE

MAE

- Preserves the same units of measurement as the data under analysis and gives all individual errors the same weights (as compared to squared error).
- This distance is easily interpretable and when aggregated over a dataset using arithmetic mean has a meaning of the average error.

MSE

- In MSE since the error being squared, any prediction error is being heavily penalized.
- Due to the square, large errors are emphasized and have a relatively greater effect on the value of the performance metric.
- The effect of relatively small errors will be even smaller.
- Penalizing extreme errors or being susceptible to outliers.

Summary

- Evaluation metrics for classification
  - Confusion matrix
  - Accuracy
  - Precision-Recall
  - F1score
  - Precision-Recall curve
  - ROC
- Evaluation metrics for regression
  - R2
  - MSE
  - MAE
  - MAPE

References

- Andreas C. Müller and Sarah Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media; 1st edition.
- Andreas C. Müller, COMS W4995 Applied Machine Learning, Columbia University, Spring 2019.
- ML501 Machine Learning, Intel AI Academy, 2017
- https://en.wikipedia.org/wiki/Precision_and_recall