Example: Decision tree on MNIST
Random forest
Example: DT, RF, KNN not very good
Basic linear classifier
Support vector machine
Underfitting and overfitting
Parameter tuning
- Cross-validation
- Machine learning pipeline
- Learning on imbalanced data
Cross Validation (CV) Algorithm

The whole dataset 100%
Cross Validation (CV) Algorithm

Training data 80%

Test 20%
Cross Validation (CV) Algorithm

Training data 80%
Cross Validation (CV) Algorithm

Training data 80%
Cross Validation (CV) Algorithm

Training data 80%

Train on 60% of data  Validate on 20%
Cross Validation (CV) Algorithm

20%  20%  20%  20%

Training data 80%

Train  Train  Train  Val

Train on 60% of data  Validate on 20%
Cross Validation (CV) Algorithm

Training data 80%

Train on 60% of data  Validate on 20%

20% 20% 20% 20% 

• → 0.75
Cross Validation (CV) Algorithm

Training data 80%

- 20% Train
- 20% Train
- 20% Train
- 20% Val

- 0.75

- 0.85
Cross Validation (CV) Algorithm

Training data 80%

- Train
- Train
- Val
- Train
- Train
- Val
- Train
- Train

- • ➤ 0.75
- • ➤ 0.85
- • ➤ 0.91
Cross Validation (CV) Algorithm

Training data 80%

1. Train → Train → Train → Val → 0.75
2. Train → Train → Val → Train → 0.85
3. Train → Val → Train → Train → 0.91
4. Val → Train → Train → Train → 0.68
Cross Validation (CV) Algorithm

Training data 80%

\[
\text{MEAN (0.75, 0.85, 0.91, 0.68) = ?}
\]
Cross Validation (CV) Algorithm

Training data 80%

MEAN (0.75, 0.85, 0.91, 0.68) = 0.75
Cross Validation (CV) Algorithm

Training data 80%

Choose the **best** model/parameters based on this estimate and then apply it to the test set.

\[
\text{MEAN (0.75, 0.85, 0.91, 0.68) = 0.75}
\]
The aim of cross-validation is:

A. Learning better models
B. Evaluating the models
C. Optimising parameters of learning algorithms
D. Not sure

[Bar chart showing response distribution:]
- Learning better models: 1
- Evaluating the models: 1
- Optimising parameters of learning algorithms: 1
- Not sure: 1
Example: Decision tree on MNIST
Random forest
Example: DT, RF, KNN not very good
Basic linear classifier
Support vector machine
Underfitting and overfitting
Parameter tuning
Cross-validation
- Machine learning pipeline
- Learning on imbalanced data
Machine Learning pipeline

Raw Data
Machine Learning pipeline

Raw Data ➔ Preprocessing
Machine Learning pipeline

Raw Data $\rightarrow$ Preprocessing $\rightarrow$ Feature extraction
Machine Learning pipeline

Raw Data ➔ Preprocessing ➔ Feature extraction

Split into train & test ➔ test set
Machine Learning pipeline

Raw Data → Preprocessing → Feature extraction

Choose a model → Split into train & test → test set
Machine Learning pipeline

Raw Data → Preprocessing → Feature extraction

Choose a model ← Split into train & test

Find best parameters using CV ← test set
Machine Learning pipeline

Raw Data ➔ Preprocessing ➔ Feature extraction

Find best parameters using CV

Choose a model

Split into train & test

test set
Machine Learning pipeline

Raw Data ➔ Preprocessing ➔ Feature extraction

Find best parameters using CV ➔ Choose a model ➔ Split into train & test

Train the model on the whole training set ➔ test set
Machine Learning pipeline

Raw Data → Preprocessing → Feature extraction

Find best parameters using CV

Choose a model

Split into train & test

Train the model on the whole training set

Evaluate final model on the test set
Machine Learning pipeline

Raw Data → Preprocessing → Feature extraction

Find best parameters using CV → Choose a model → Split into train & test

Train the model on the whole training set → Evaluate final model on the test set

Report your results
Machine Learning pipeline

1. Raw Data
2. Preprocessing
3. Feature extraction
4. Choose a model
5. Find best parameters using CV
6. Split into train & test
7. Train the model on the whole training set
8. Evaluate final model on the test set
9. Report your results
Lecture 07 – Machine learning 2

✓ Example: Decision tree on MNIST
✓ Random forest
✓ Example: DT, RF, KNN not very good
✓ Basic linear classifier
✓ Support vector machine
✓ Underfitting and overfitting
✓ Parameter tuning
✓ Cross-validation
✓ Machine learning pipeline

• Learning on imbalanced data
What if your data is imbalanced?
What if your data is imbalanced?
What if your data is imbalanced?
What if your data is imbalanced?

Majority class classifier has accuracy = 92%

Accuracy is not the most suitable evaluation measure here
Examples of imbalanced tasks

• Internet search:
  – Few relevant pages (positive class)
  – Many irrelevant pages (negative class)

• Medical diagnostic testing
  – Few disease cases (positive class)
  – Many healthy cases (negative class)
# Confusion matrix

(a.k.a. contingency table)

<table>
<thead>
<tr>
<th>Actual = Yes</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positives (TP)</td>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False positives (FP) (Type I error)</td>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted positives (PPos)</th>
<th>Predicted negatives (PNeg)</th>
<th>Pred +</th>
<th>Pred -</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>-</td>
<td>30</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
<td>60</td>
<td>100</td>
</tr>
</tbody>
</table>

Example
False Positive

Type I Error

You’re pregnant!

False Negative

Type II Error

You’re not pregnant!
# Evaluation measures

<table>
<thead>
<tr>
<th>Actual = Yes</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual = No</td>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted positives (PPos)</td>
<td>Predicted negatives (PNeg)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example**

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>- 30</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>100</td>
</tr>
</tbody>
</table>
## Evaluation measures

<table>
<thead>
<tr>
<th></th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual = Yes</td>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
</tr>
<tr>
<td>Actual = No</td>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Pred +</th>
<th>Pred -</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positives (Pos)</td>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Negatives (Neg)</td>
<td>30</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>Predicted positives (PPos)</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Predicted negatives (PNeg)</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

\[
\text{Accuracy} = \frac{(TP + TN)}{Total} = \frac{(10 + 50)}{100} = 0.60
\]
Evaluation measures

<table>
<thead>
<tr>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual = Yes</td>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual = No</td>
<td>False positives (FP) (Type I error)</td>
<td>True negatives (TN)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Predicted positives (PPos)  Predicted negatives (PNeg)

Example

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
<td>60</td>
</tr>
</tbody>
</table>

**Accuracy** = \( \frac{TP + TN}{Total} \)

**Precision** = \( TP / PPos \)

\[ \text{Accuracy} = \frac{10 + 50}{100} = 0.60 \]

\[ \text{Precision} = \frac{10}{40} = 0.25 \]
# Evaluation measures

<table>
<thead>
<tr>
<th>Actual = Yes</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
<th>Actual = No</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
<td></td>
<td>False positives (FP) (Type I error)</td>
</tr>
<tr>
<td>True negatives (TN)</td>
<td></td>
<td></td>
<td>True negatives (TN)</td>
</tr>
<tr>
<td>Predicted positives (PPos)</td>
<td>Predicted negatives (PNeg)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Positives (Pos)</th>
<th>Negatives (Neg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

**Accuracy** = \( \frac{TP + TN}{Total} \)

**Precision** = \( \frac{TP}{PPos} \)

**Recall** = \( \frac{TP}{Pos} \)

\[
\text{Accuracy} = \frac{10 + 50}{100} = 0.60 \\
\text{Precision} = \frac{10}{40} = 0.25 \\
\text{Recall} = \frac{10}{20} = 0.50
\]
Evaluation measures

<table>
<thead>
<tr>
<th>Actual = Yes</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positives (TP)</td>
<td>False negatives (FN) (Type II error)</td>
<td>Positives (Pos)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actual = No</th>
<th>Predicted = Yes</th>
<th>Predicted = No</th>
</tr>
</thead>
<tbody>
<tr>
<td>False positives (Type I error)</td>
<td>Predicted positives (PPos)</td>
<td>Predicted negatives (PNeg)</td>
</tr>
</tbody>
</table>

Accuracy = (TP + TN) / (Pos + Neg)

Precision = TP / PP

Recall = TP / Pos

\[ F - \text{measure} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \]

For imbalanced datasets, maximizing F-measure is better than maximizing Accuracy.

For example, majority class classifier has F-measure = 0

Example

<table>
<thead>
<tr>
<th>Pred +</th>
<th>Pred -</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 10</td>
<td>10</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>60</td>
<td>100</td>
</tr>
</tbody>
</table>

\[ \text{Recall} = 10 \div 20 = 0.50 \]

\[ F - \text{measure} = \frac{2}{\frac{1}{0.25} + \frac{1}{0.50}} = 0.40 \]
How to learn on imbalanced data?

1. Undersampling

8%  92%
How to learn on imbalanced data?

Get rid of as many instances of dominant class as needed to roughly balance the dataset.

40% 60%
How to learn on imbalanced data?

1. Undersampling

Get rid of as many instances of dominant class as needed to roughly balance the dataset.

40% 60%

We lose a lot of data
How to learn on imbalanced data?

1. Undersampling

Get rid of as many instances of dominant class as needed to roughly balance the dataset.

Instances could be called samples or observations.

We lose a lot of data.
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
How to learn on imbalanced data?

1. Undersampling

2. Oversampling

Make **multiple copies** of instances of the smaller class so that it would roughly balance the dataset

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

23% 73%
How to learn on imbalanced data?

1. Undersampling
2. Oversampling

You need to make new images a bit different from the old ones.

Make multiple copies of instances of the smaller class so that it would roughly balance the dataset.

This does not always help.
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
3. Augmentation
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
3. Augmentation
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
3. Augmentation

Unfortunately is not so easy for other data domains
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
3. Augmentation
How to learn on imbalanced data?

1. Undersampling
2. Oversampling
3. Augmentation
4. Adequate performance estimate (for example F-measure)
✓ Example: Decision tree on MNIST
✓ Random forest
✓ Example: DT, RF, KNN not very good
✓ Basic linear classifier
✓ Support vector machine
✓ Underfitting and overfitting
✓ Parameter tuning
✓ Cross-validation
✓ Machine learning pipeline
✓ Learning on imbalanced data