Practical Considerations

Machine Learning and Computational Statistics (DSC6135)

Instructors: Weiwei Pan (Harvard), Javier Zazo (Harvard), Melanie F. Pradier (Harvard)

So far...

we have seen multiple approaches for regression and classification tasks in machine learning.

Regardless of which model you use, the machine learning pipeline is always the same (`fit` and then `predict`).

Example: Regression

In [ ]:

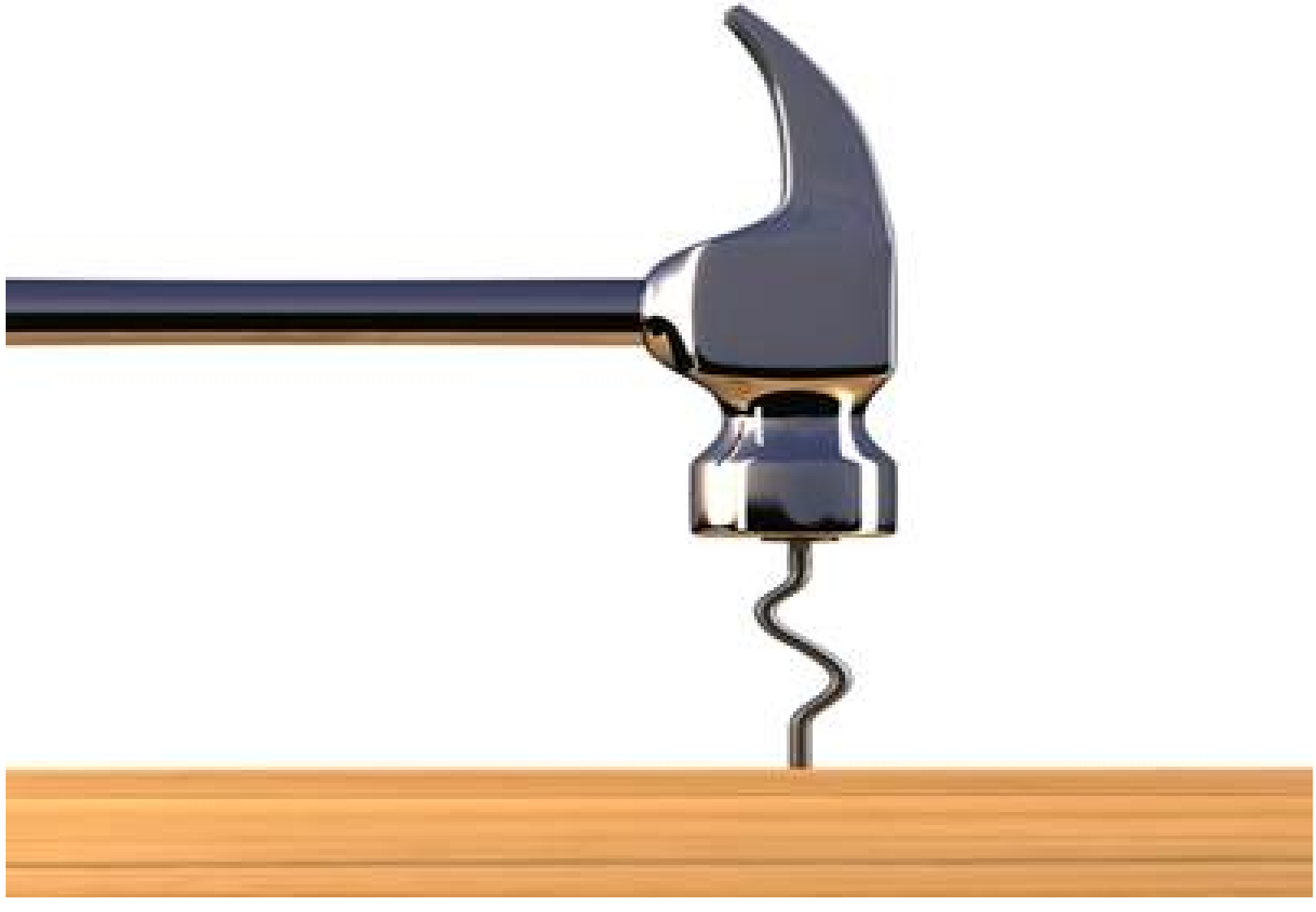
```
# given some data...
x_train, x_test, y_train, y_test = get_data()
# customize your model
my_model = LinearRegression()
# train your model
my_model.fit(x_train, y_train)
# make predictions for new observations
y_test = my_model.predict(x_test)
# test your model
my_model.score(x_test, y_test)
```

Example: Classification

In [ ]:

```
# given some data...
x_train, x_test, y_train, y_test = get_data()
# customize your model
my_model = LogisticRegression(solver='lbfgs')
# train your model
my_model.fit(x_train, y_train)
# make predictions for new observations
y_test = my_model.predict(x_test)
# test your model
my_model.score(x_test, y_test)
```


Hi-Spec
#0
#1
2.4mm
3.0mm
Hi-Spec
Hi-Spec
12 ft. (3.5 m)
Tape Measure
Made in China



Some practical considerations

1. How to deal with other kinds of data?
2. How to improve quality of features?
3. How to inspect/understand your model?
4. How to evaluate your classifier?

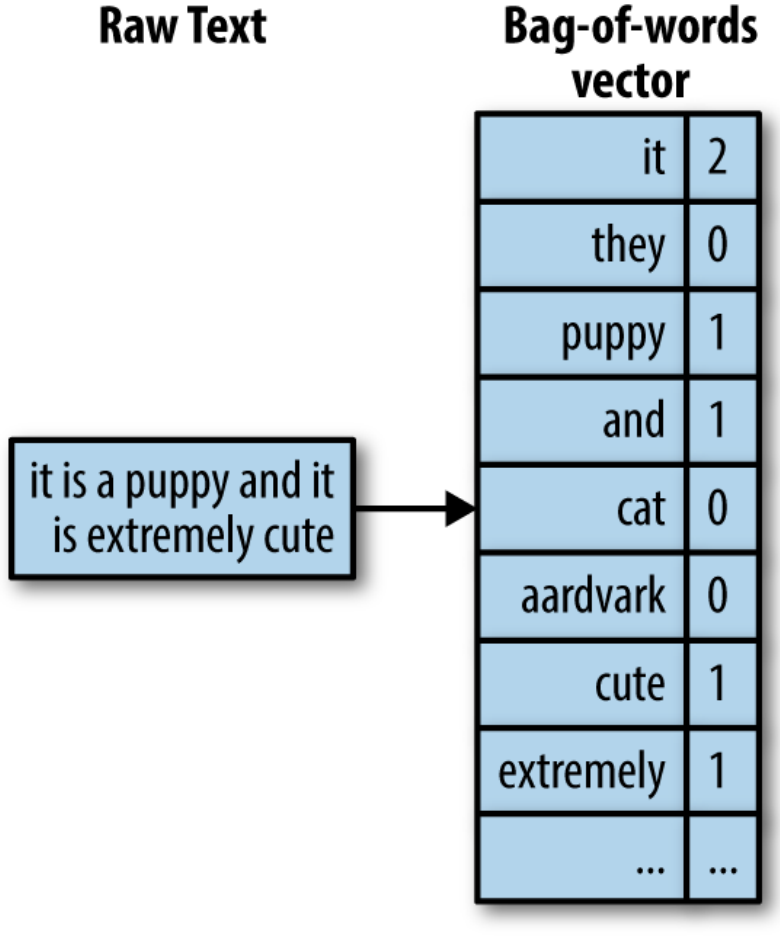
1. How to deal with other kinds of data?

1.1. Categorical Data: ”one-hot vector encoding”

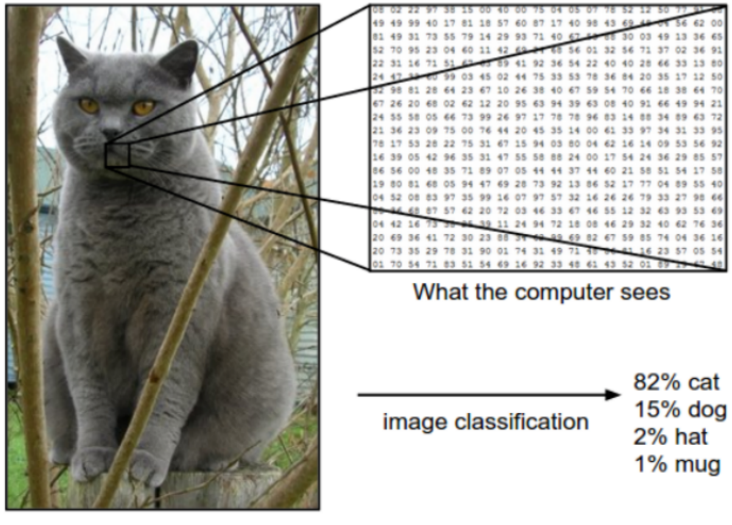
Sample	Category	Numerical
1	Human	1
2	Human	1
3	Penguin	2
4	Octopus	3
5	Alien	4
6	Octopus	3
7	Alien	4

Sample	Human	Penguin	Octopus	Alien
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	1	0
7	0	0	0	1

1.2. Text Data: ”bag-of-words”

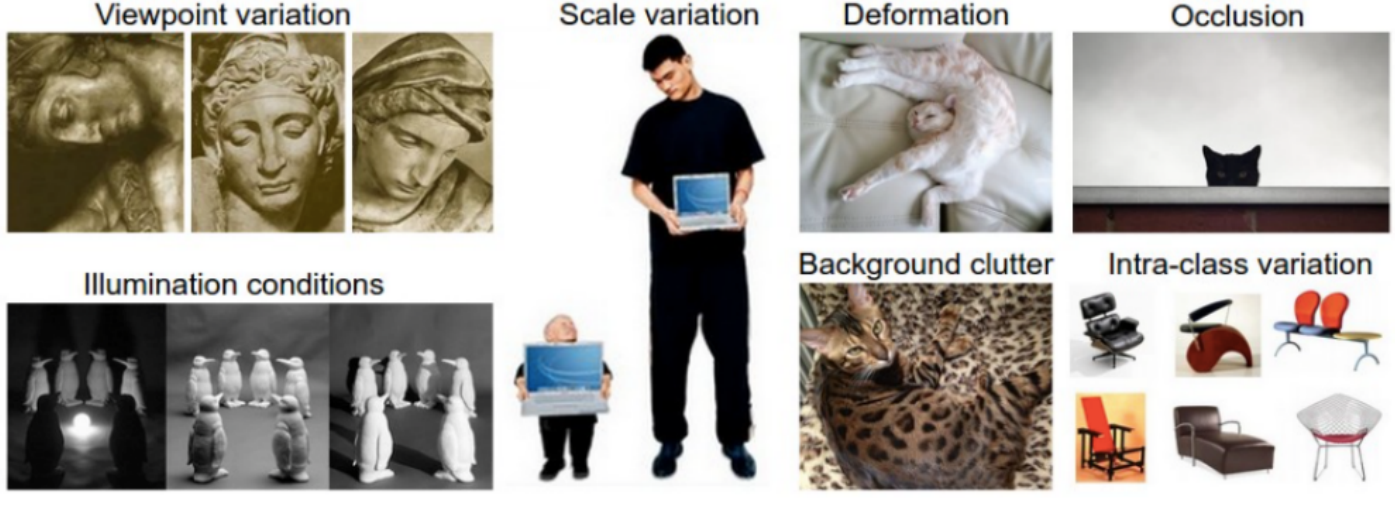


1.3. Image Data:



- Black and white: concatenation of numbers
- Colors: RGB representation ('red','green','blue')

1.3. Image Data:



2. How to improve quality of features?

We can:

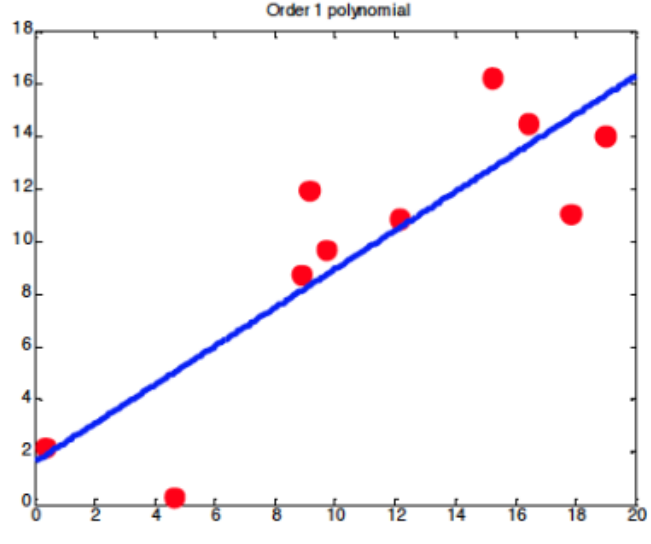
a) transform features

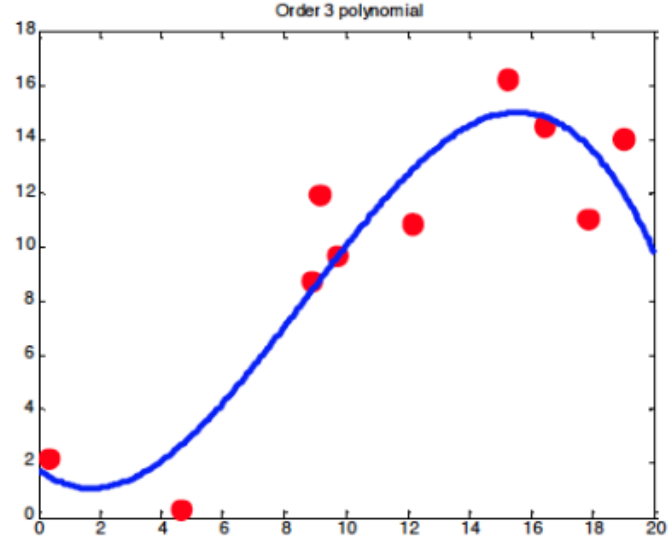
b) select features

2.1 Feature Transformation

$$x_{\text{transformed}} = \phi(x)$$

Example: linear regression with polynomial features

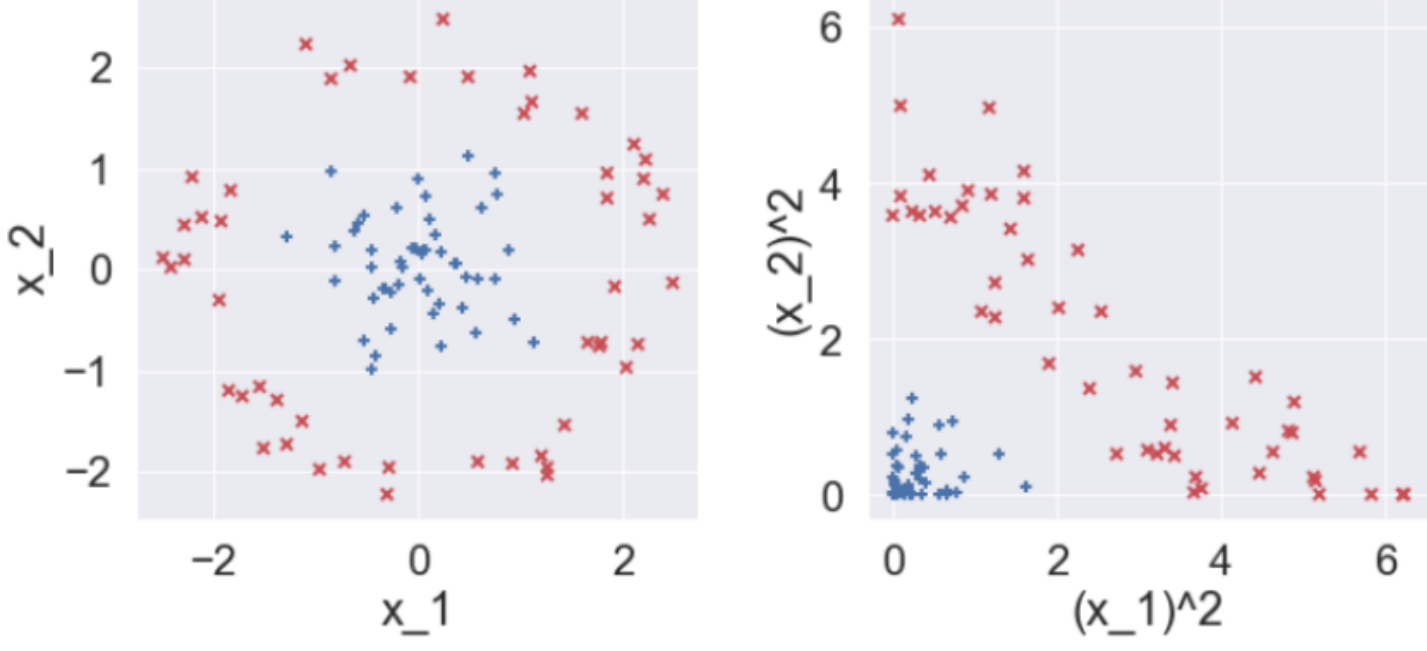




2.1 Feature Transformation

$$x_{\text{transformed}} = \phi(x)$$

Example: logistic regression with polynomial features



2.1 Feature Transformation

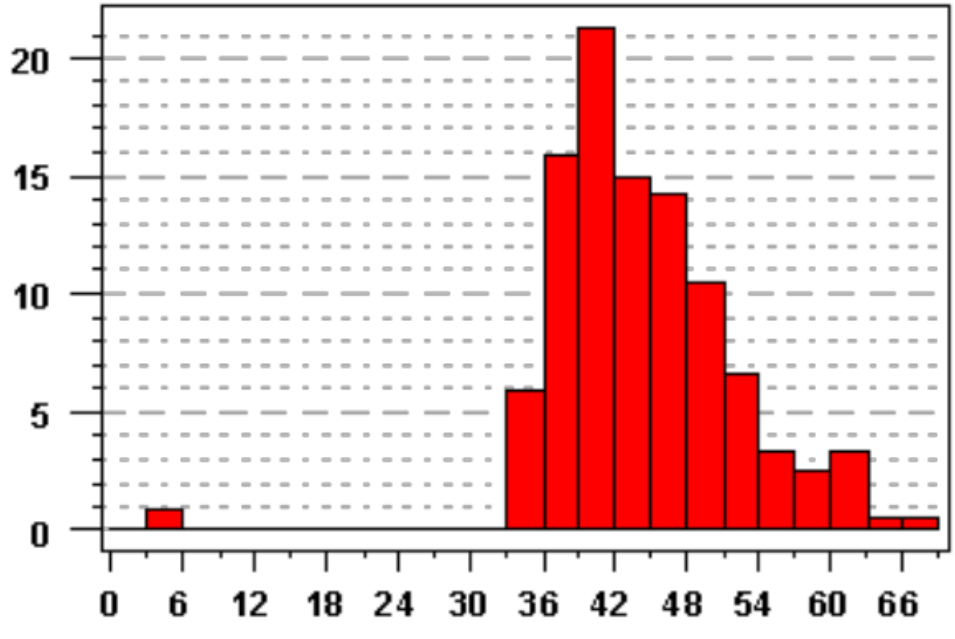
- Feature rescaling

$$\phi(x_i) = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

- Feature standardization

$$\phi(x_i) = \frac{x_i - \mu_i}{\sigma_i}$$

Question: what would happen if you standardize a feature with an outlier?



Feature scaling with outliers

sklearn.preprocessing.RobustScaler

class sklearn.preprocessing. **RobustScaler** (*with_centering=True, with_scaling=True, quantile_range=(25.0, 75.0), copy=True*) [source]

Scale features using statistics that are robust to outliers.

This Scaler removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

2.2. Feature Selection

- Based on data exploration (e.g., colinearity)
- Forward selection
- Backward selection
- L1 regularization

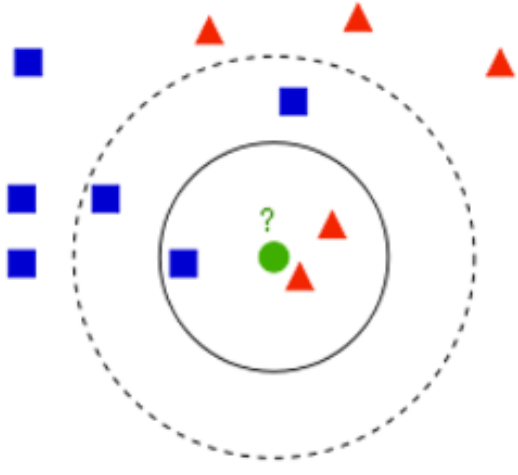
Extra challenge: what if my features have missing data?

	seq_id	hubway_id	status	duration	start_date	strt_statn	end_date	end_statn
count	1.579025e+06	1.579025e+06	1579025	1.579025e+06	1579025	1.579011e+06	1579025	1.578980e+06
unique	NaN	NaN	1	NaN	521432	NaN	515102	NaN
top	NaN	NaN	Closed	NaN	7/30/2013 17:18:00	NaN	10/19/2013 16:26:00	NaN
freq	NaN	NaN	1579025	NaN	25	NaN	27	NaN
mean	7.895130e+05	8.865317e+05	NaN	1.200280e+03	NaN	5.438039e+01	NaN	5.425603e+01
std	4.558254e+05	5.064783e+05	NaN	2.653539e+04	NaN	3.364295e+01	NaN	3.347219e+01
min	1.000000e+00	8.000000e+00	NaN	-6.900000e+03	NaN	3.000000e+00	NaN	3.000000e+00
25%	3.947570e+05	4.465250e+05	NaN	4.120000e+02	NaN	2.700000e+01	NaN	2.900000e+01
50%	7.895130e+05	8.950440e+05	NaN	6.600000e+02	NaN	4.800000e+01	NaN	4.800000e+01
75%	1.184269e+06	1.328083e+06	NaN	1.082000e+03	NaN	7.400000e+01	NaN	7.400000e+01
max	1.579025e+06	1.748022e+06	NaN	1.199446e+07	NaN	1.450000e+02	NaN	1.450000e+02

What people usually do...

- drop features with two many missing
- value imputation (example: 0-value, or mean-value)
- model-based imputation

Model-based inputation (example with k-nearest neighbors)



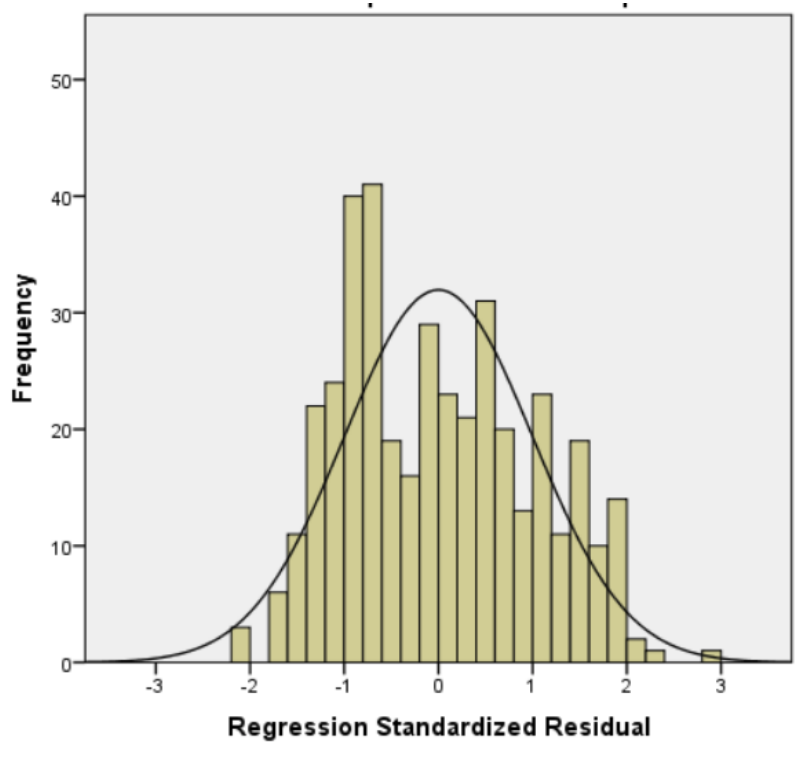
Triangles = “Female”
Squares = “Male”

3. How to inspect your model?

- Check model assumptions
- Identify most predictive features
- Check model robustness

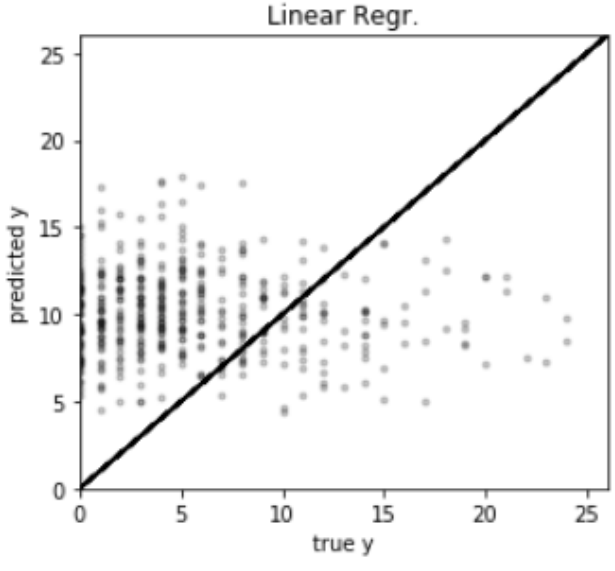
3.1. Check model assumptions

$$\epsilon = y - \hat{y}$$



3.1. Check model assumptions

$$\epsilon = y - \hat{y}$$



3.2. Identify most predictive features

Look at `feature importances`:

- For linear and logistic regression, these are the size of the coefficients
- For other models: monitor metric degradation when each feature is perturbed

3.3. Check model robustness

Sensitivity Analysis

- If the train set change, how much do the **parameters** vary?
    - confidence intervals

- If the train set change, how much do the **predictions** vary?
    - predictive intervals

Let's see it in practice!

How to evaluate your classifier?

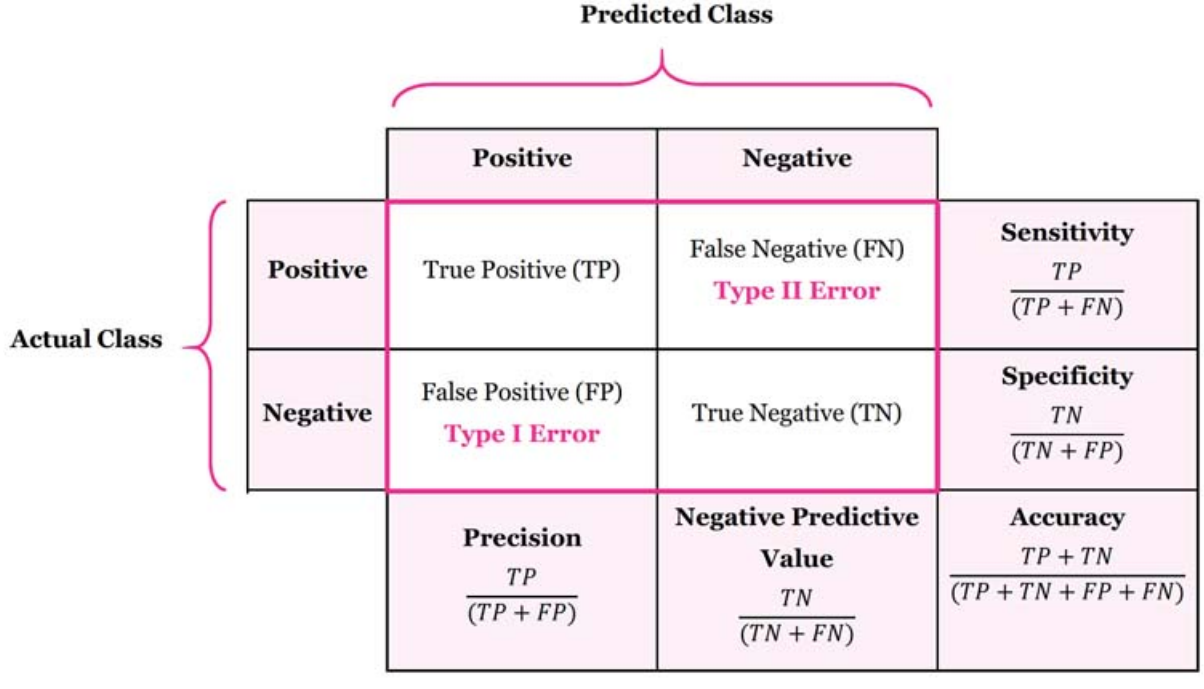
Confusion Matrix: counting mistakes in binary predictions

The following table is called the **confusion matrix**.

TN : true negative
FN : false negative
FP : false positive
TP : true positive

		Predicted label	
		Healthy (ypred=0)	Cancer (ypred=1)
True label	Healthy (y=0)	TN	FP
	Cancer (y=1)	FN	TP

Confusion Matrix: counting mistakes in binary predictions



- Sensitivity = Recall = "True positive rate (TPR)"
- Specificity = 1 - False positive rate (FPR) = True Negative Rate (TNR)
- Precision = Positive Predictive Value (PPV)

Other evaluation metrics:

- **Accuracy**: fraction of correct predictions

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

- **F1-score**: fraction of correct predictions

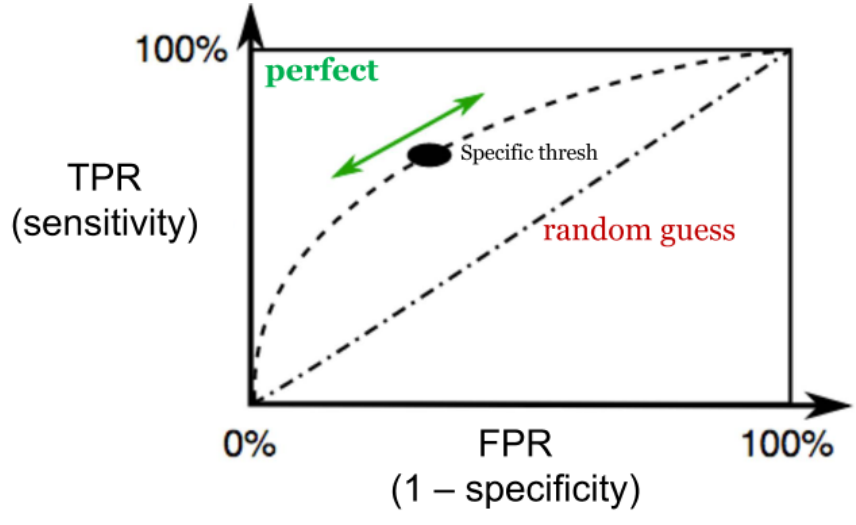
$$\text{F1-score} \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Note: Accuracy useless if classes are unbalanced!

Emphasis: we should choose the metric based on the application

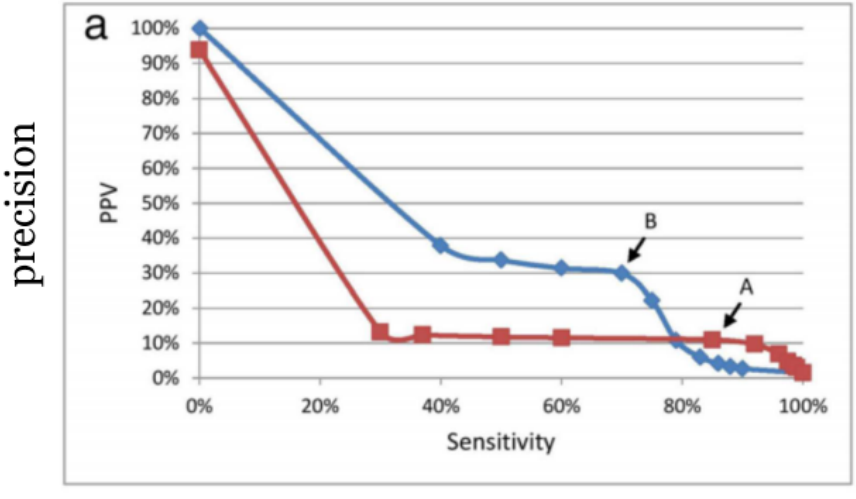
Let's see some case studies!

ROC Curve (across thresholds)



- Area under the Curve (AUROC or AUC or `C statistic`)

Precision-Recall Curve



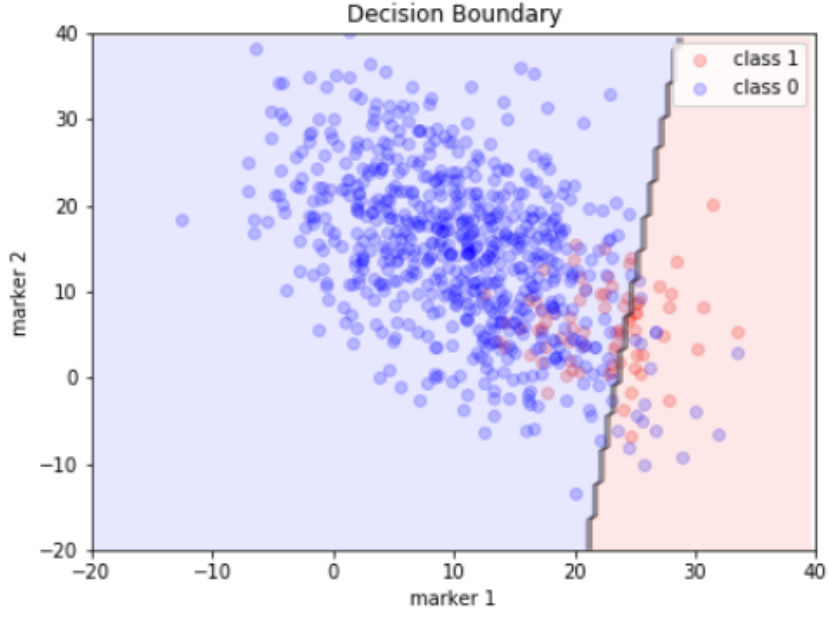
precision

recall (= TPR)

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

Some examples: `metrics.precision_score`, `metrics.recall_score`, `metrics.average_precision_score`, `metrics.roc_auc_score`

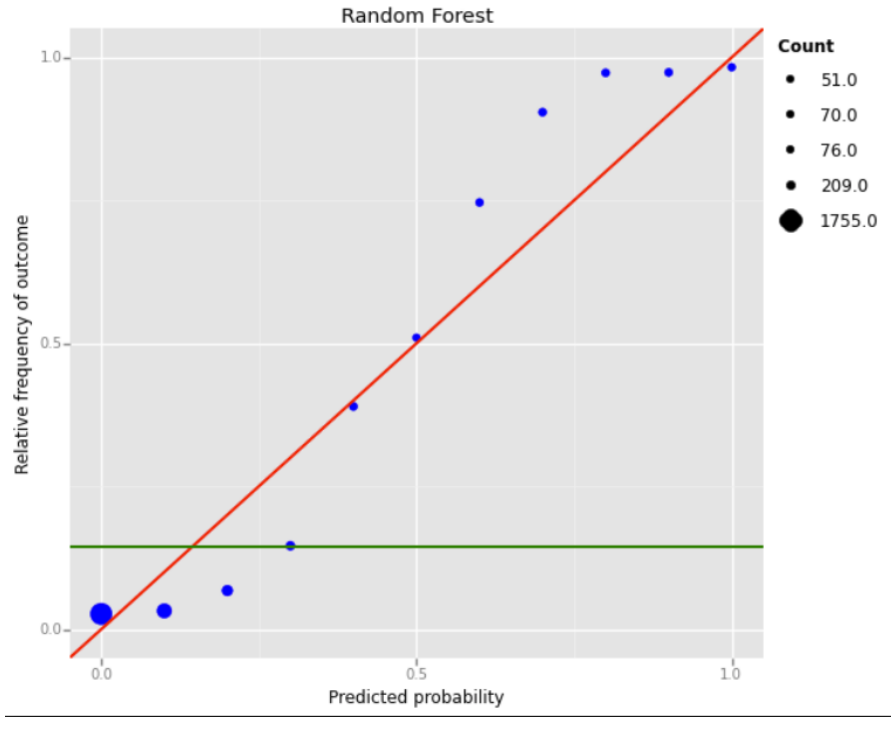
How to fix the problem of unbalanced classes?



- Upsampling
- Downsampling
- Reweighting (modify loss function)

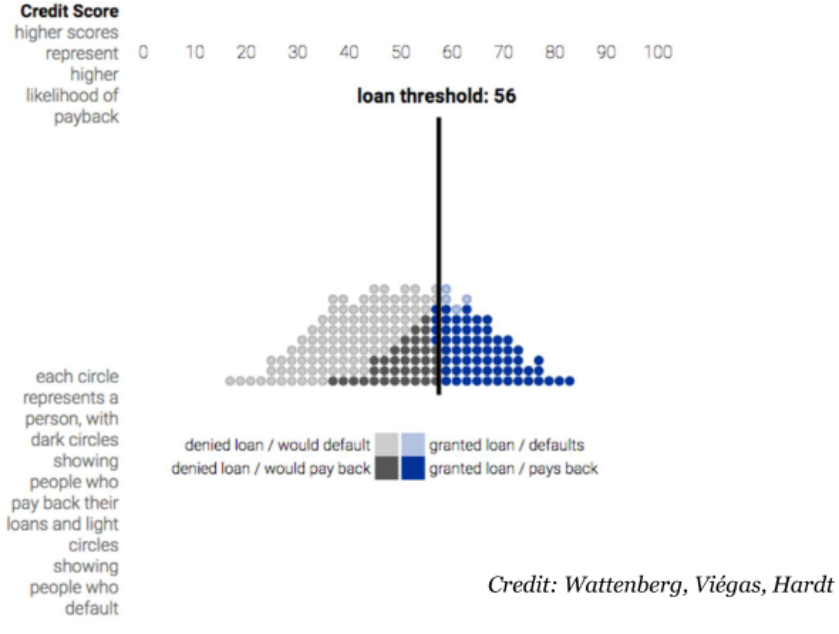
Calibration

Question: Is the `random forest` classifier conservative or adventurous in its predictions?



Exercise: Thresholding to get Binary Decisions

- Interactive Demo: [https://research.google.com/bigpicture/attacking-discrimination-in-ml/ (https://research.google.com/bigpicture/attacking-discrimination-in-ml/)](https://research.google.com/bigpicture/attacking-discrimination-in-ml/)



Credit: Wattenberg, Viégas, Hardt

Goals:

- What threshold maximizes accuracy?
- What threshold maximizes profit?