Daniele Lunghi

Machine Learning for credit card fraud detection

Presentation Outline

Introduce the basics (20 mins)

Technical part (20 mins)

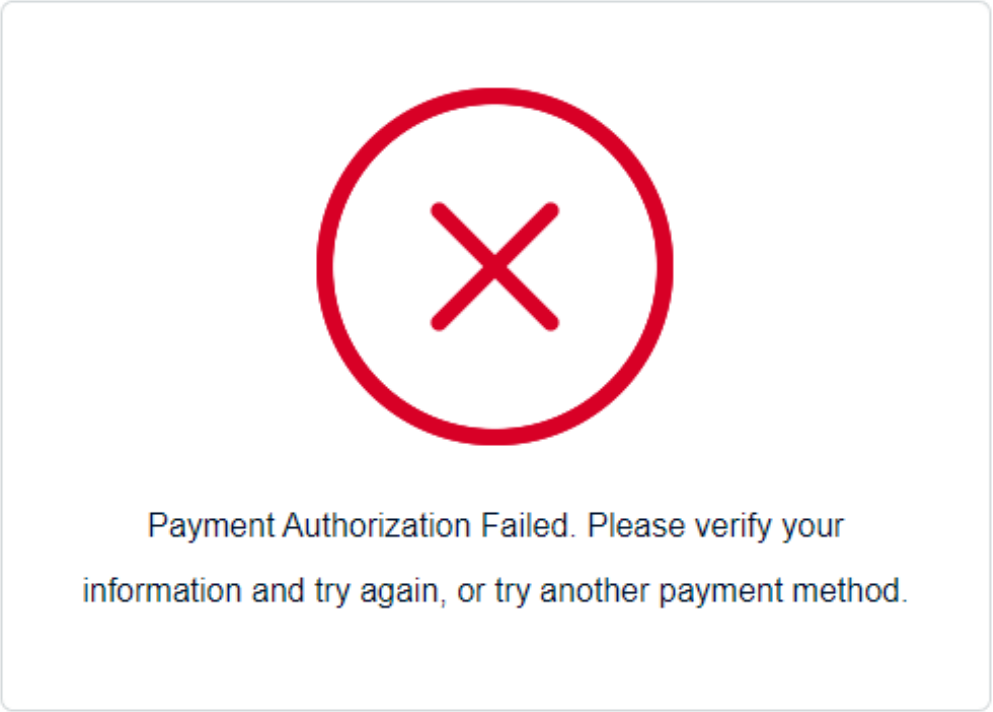
Future challenges (my work ☺) 10 mins

Scenario 1: A genuine transaction

- You are about to make your payment
- The payment goes well
- What happened?

Scenario 2: A Detected Fraud

- Someone clones your card
- They connect to Steam and try to buy a new game
- Sorry, we need to authorize
- What happened?

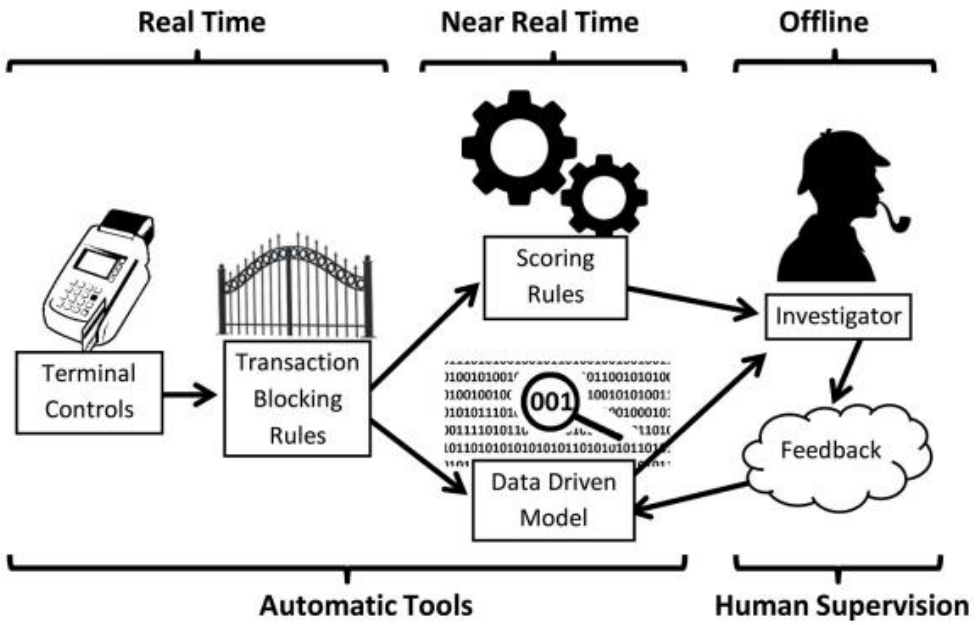


What happened

- In both cases, someone made a transaction
- “Something” decided whether a transaction was a fraud
- What is this something?

Real-world fraud detection

- Combination of humans and automated tools
- Simple rules and complex machine learning combined
- WE FOCUS ON MACHINE LEARNING



Classification

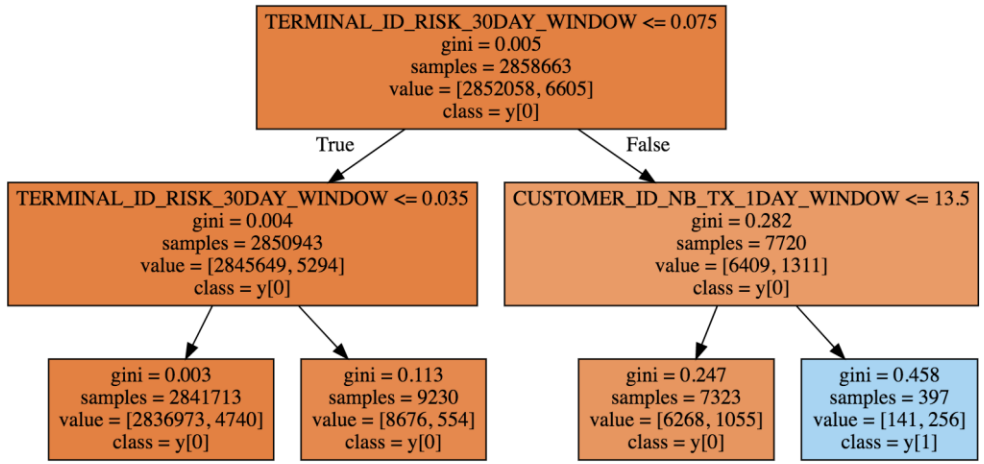
- Automatic method predict the correct label of a given input data
- Rule-based classification:
  - IF AMOUNT > $200K SIGNAL AS FRAUD
  - IF AMOUNT > $400 AND CATEGORY == VIDEOGAME, SIGNAL AS FRAUD

Machine Learning Classification

- *How many of you know something about machine learning?*
- "Classification algorithms used in machine learning utilize input training data for the purpose of predicting the likelihood or probability that the data that follows will fall into one of the predetermined categories" *(The Internet)*

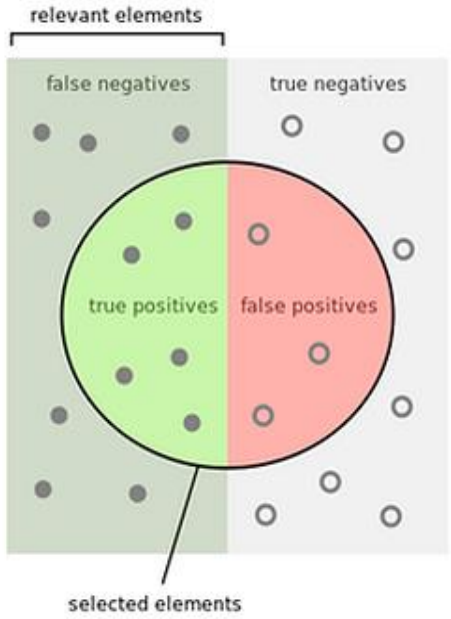
Decision Tree

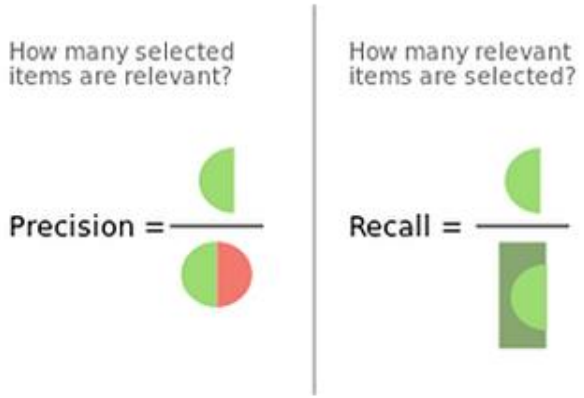
- Data are split based on a condition
- Hierarchical splits form a tree
- Ultimate goal: Separate classes with best accuracy and minimal complexity



Metrics

- Source: *The Internet*





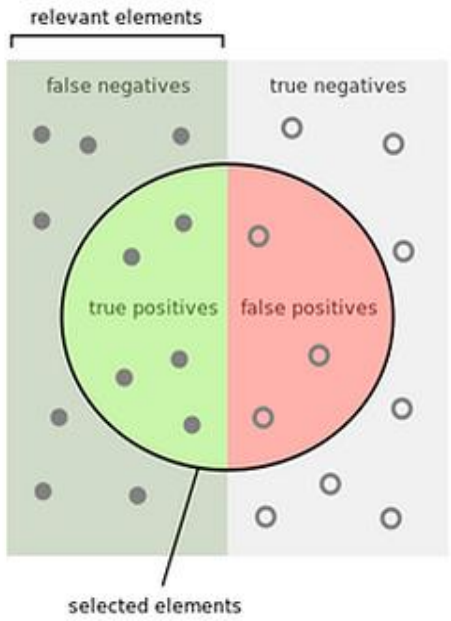
Metrics

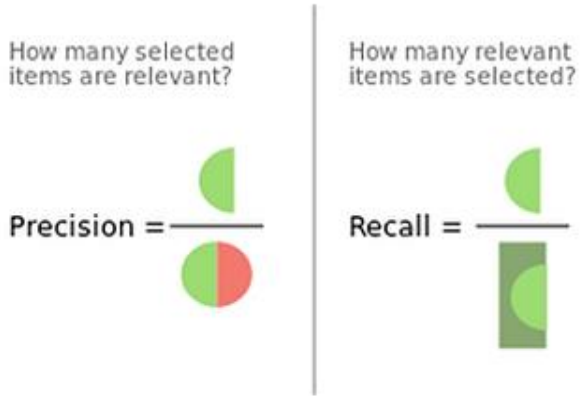
- Accuracy = $\frac{TN+TP}{TN+TP+FN+FP}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1 Score = $\frac{2*Precision*Recall}{Precision+Recall}$

TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

Metrics

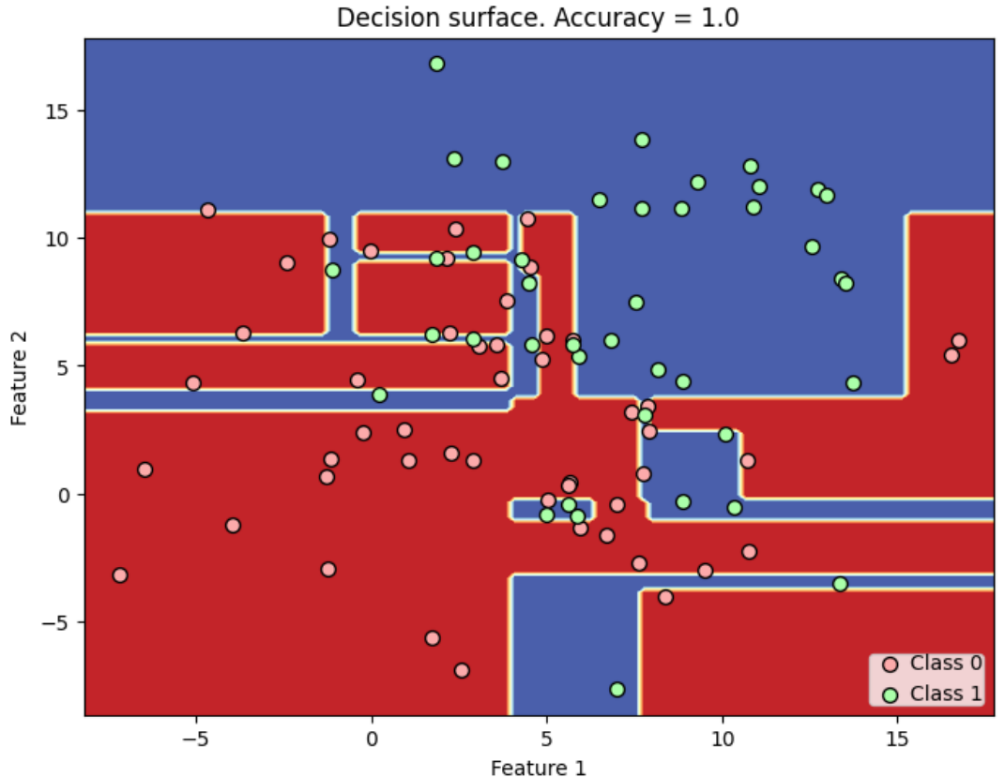
- Source: *The Internet*





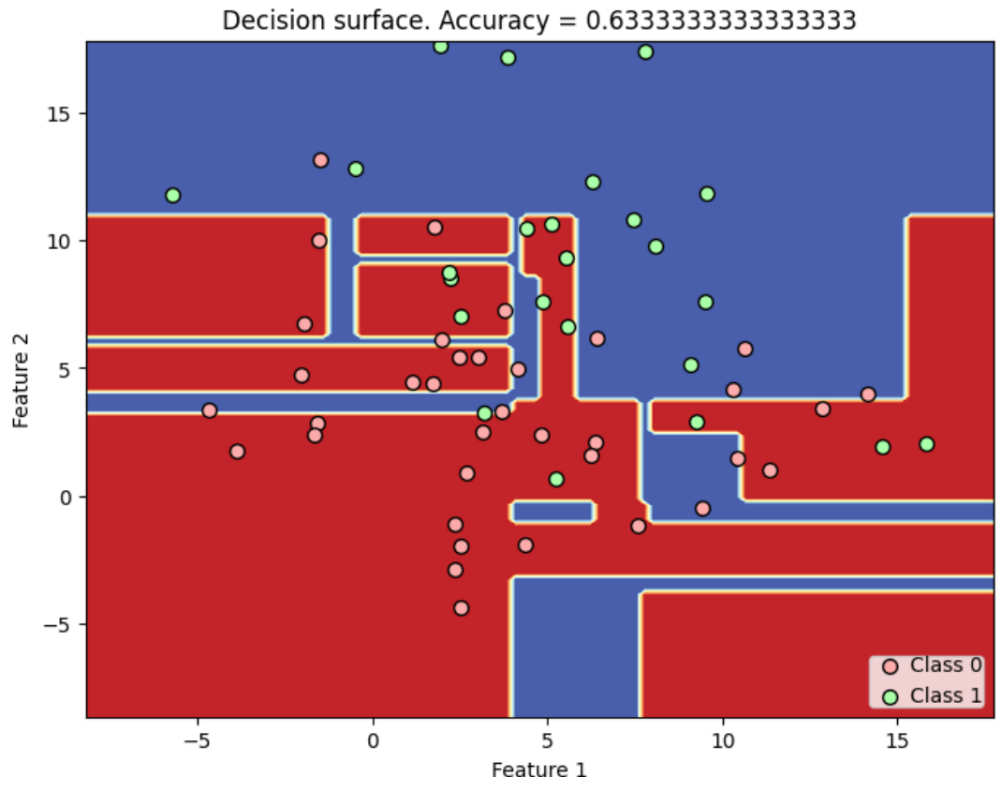
Applying our decision tree

- A perfect result (100% split)
- Decision boundaries seem a bit too complicated
- Are we sure we are understanding reality?



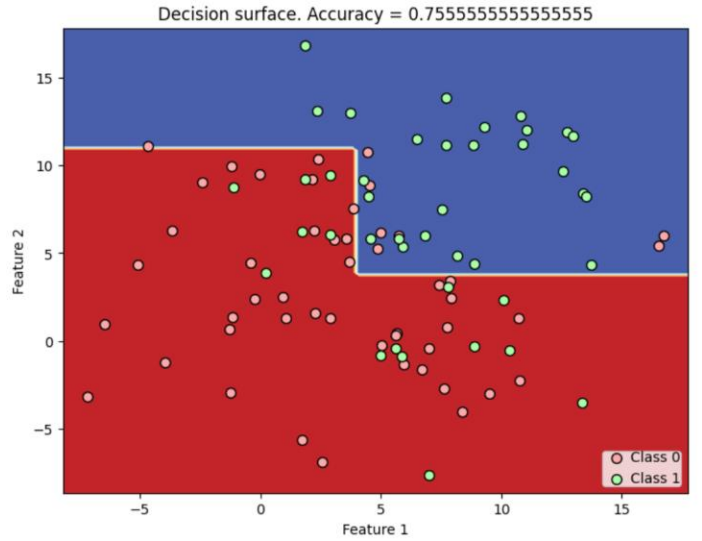
Let’s test in on more data!

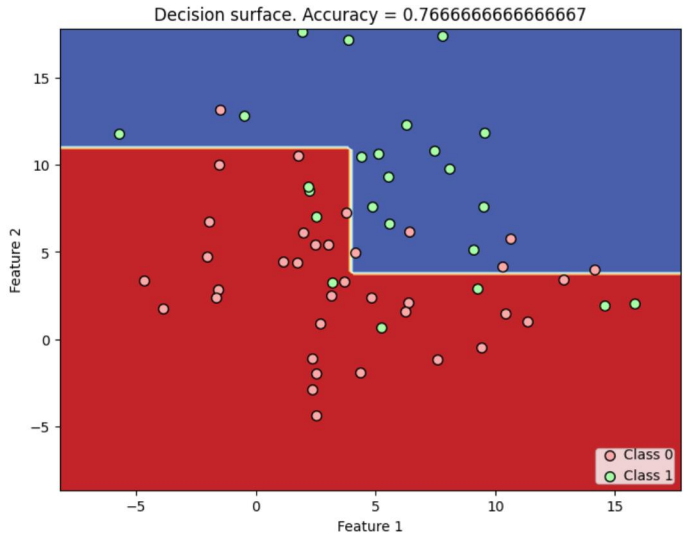
- We use other data from the same generation process and test the performance
- Test set, accuracy drops dramatically
- What is happening here?



A simpler tree

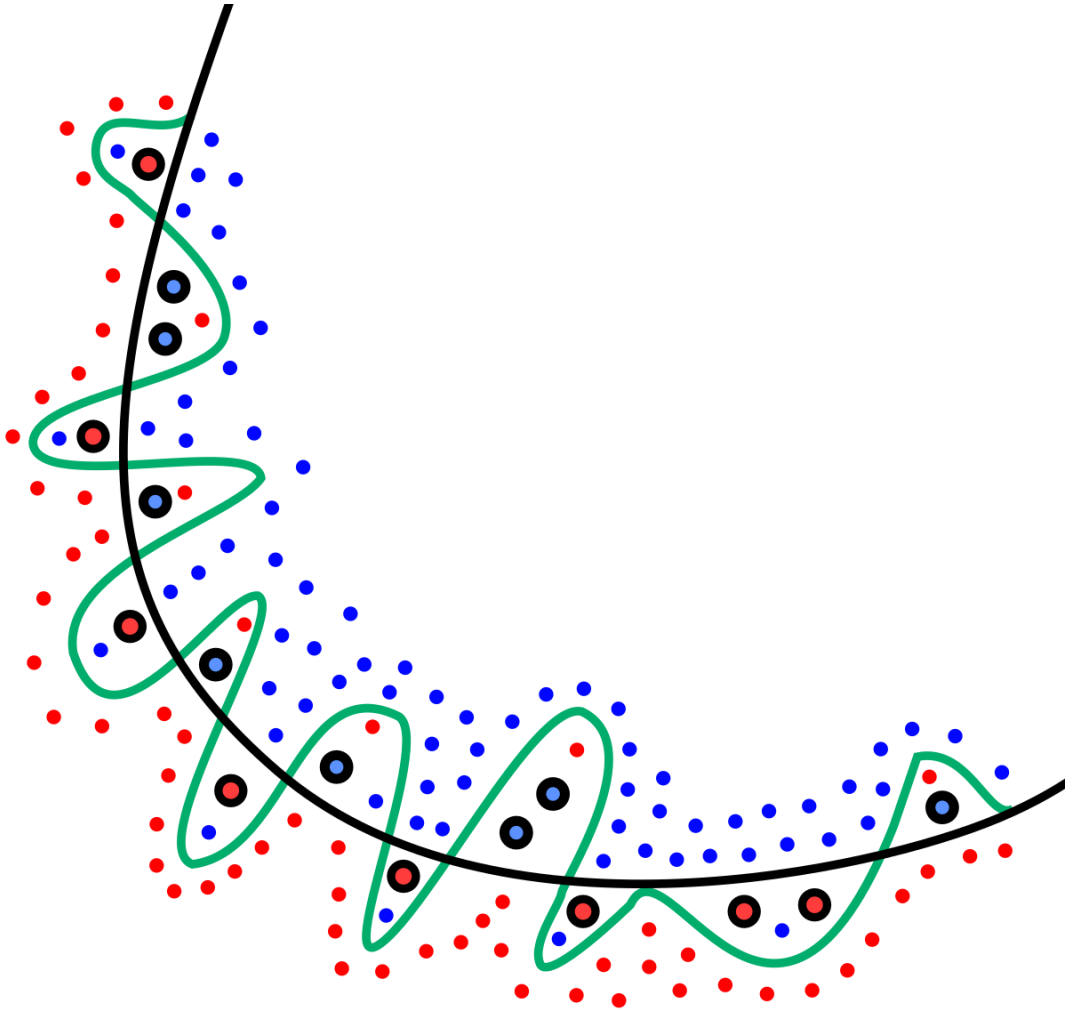
- Decision boundaries are simpler
- Accuracy on the training set is lower
- Test set accuracy is better
- We are generalizing. We are learning!





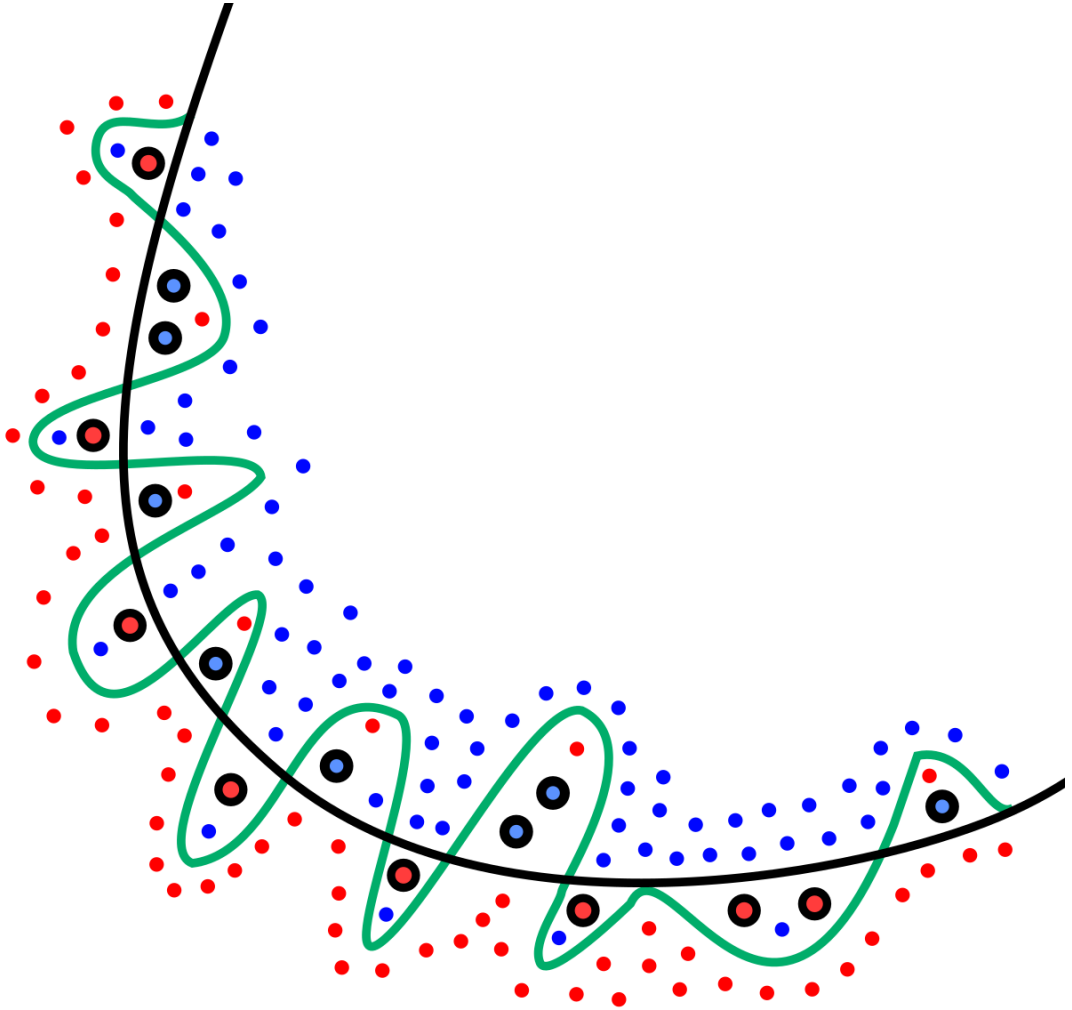
Generalization and overfitting

- Arguably, the most important concepts in machine learning
- It is not enough to perfectly divide the training set, our findings MUST GENERALIZE
- Data is noisy, when we are learning too much from noise, we are overfitting
- We need a test set to measure the effective performance of our classifier



Some more overfitting

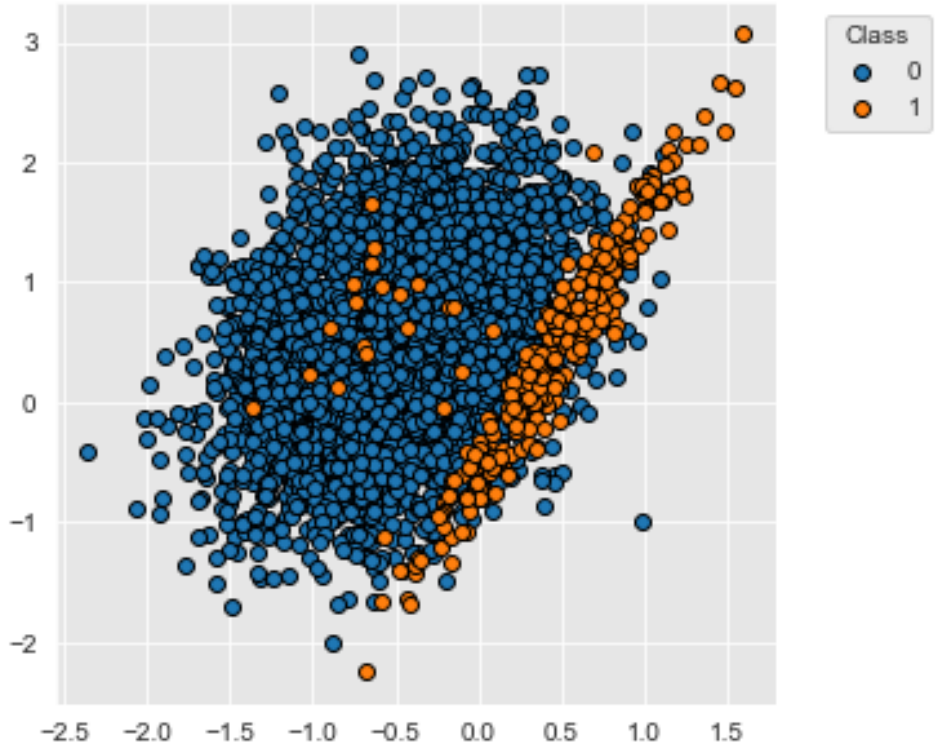
- Which line would you prefer?
- Which one will generalize better?



From machine learning to fraud detection

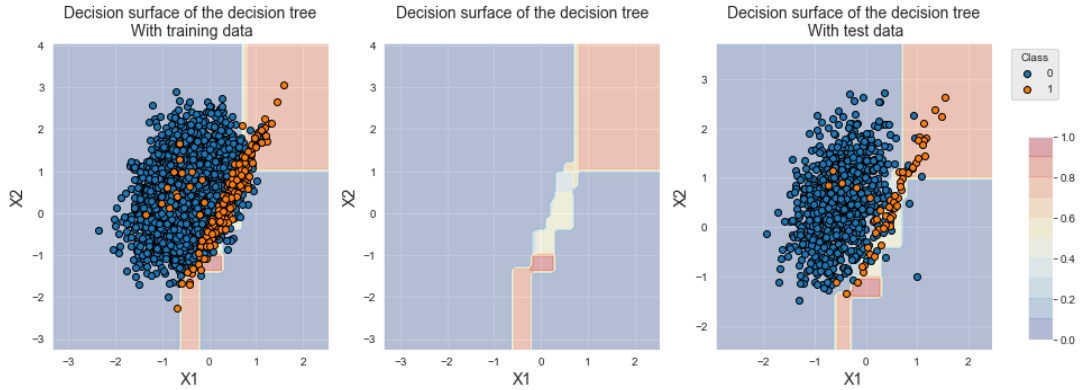
But frauds are a small minority

- Most transactions performed everyday are genuine
- Can we learn how frauds look like if we see mainly genuine transactions?
- Will we be able to generalize?



Data Imbalance

- The vast majority of the data space looks genuine to our model
- This may reflect reality
- But we accept many False Negative (FN). We miss many orange points
- These are undetected frauds!



Metrics Revisited

- *Which metrics will be affected by having a small number of positives?*

- Accuracy = $\frac{TN+TP}{TN+TP+FN+FP}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1 Score = $\frac{2*Precision*Recall}{Precision+Recall}$

TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

Metrics Revisited

- *Which metrics will be affected by having a small number of positives?*

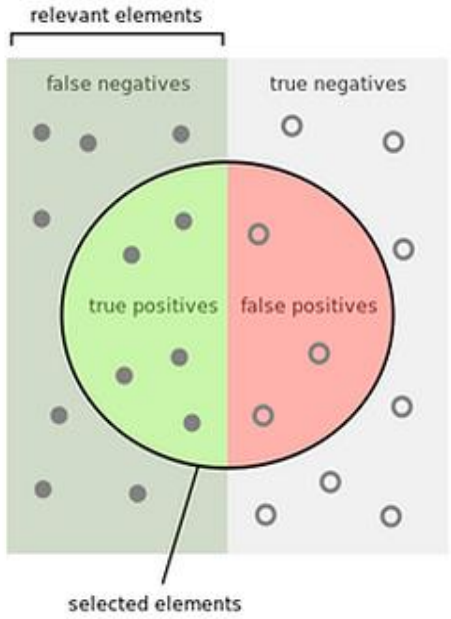
- Accuracy = $\frac{TN+TP}{TN+TP+FN+FP}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1 Score = $\frac{2*Precision*Recall}{Precision+Recall}$

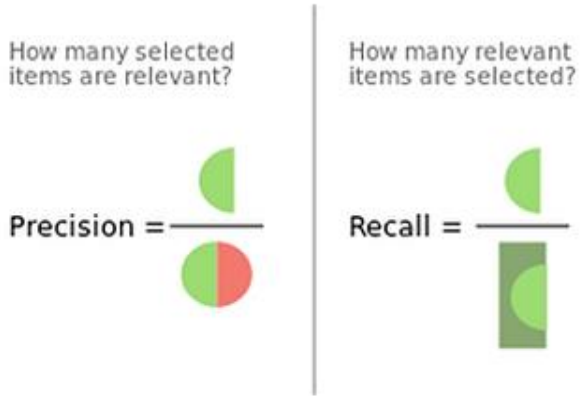
TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

Precision might actually increase!

Metrics

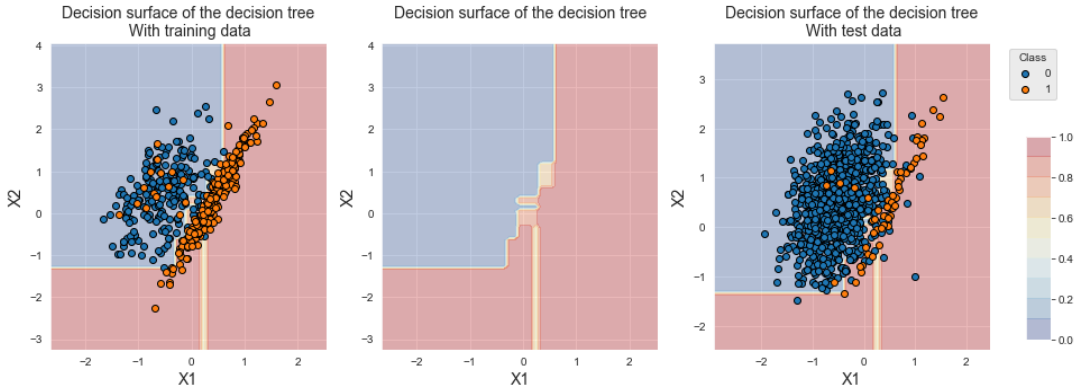
- Source: *The Internet*





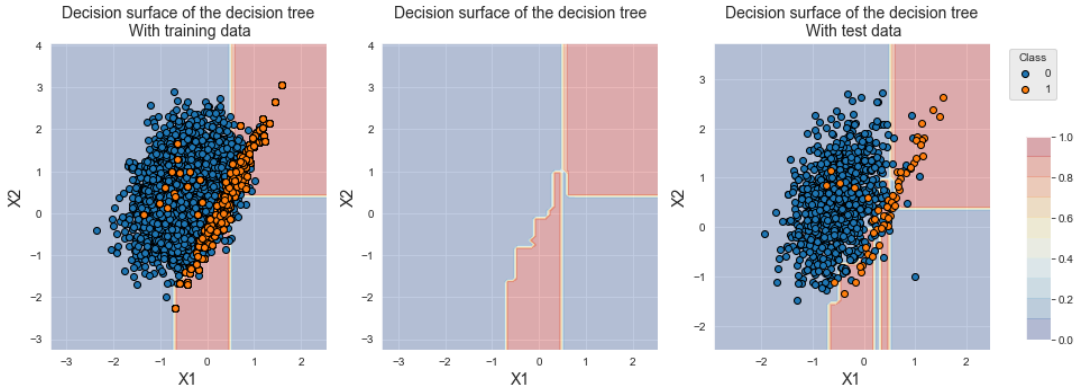
(Random) Undersampling

- Goal: rebalance the training set
- Select a random subset from the majority class
- Decisions boundary look more stable
- For who know a bit of statistics, *we are changing the model's prior*



(Random) Oversampling

- Artificially duplicate the frauds in the training set
- Intuitively, frauds now weight more, and affect the classifier
- Again, we are changing the prior



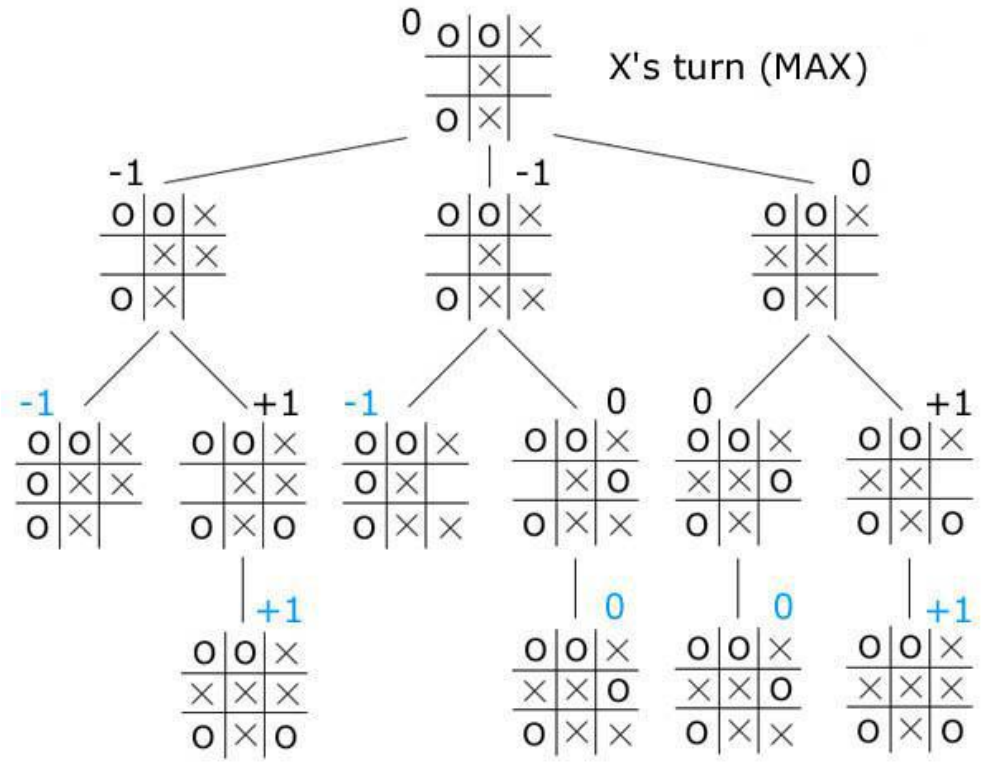
Honorable mentions

- More complex classifier: Random Forests, Neural Networks…
- More complex resampling methods: SMOTE, K-Means, GAN….
- Concept drift, delayed feedback…
- And much more

Adversarial Attacks (My research ☺)

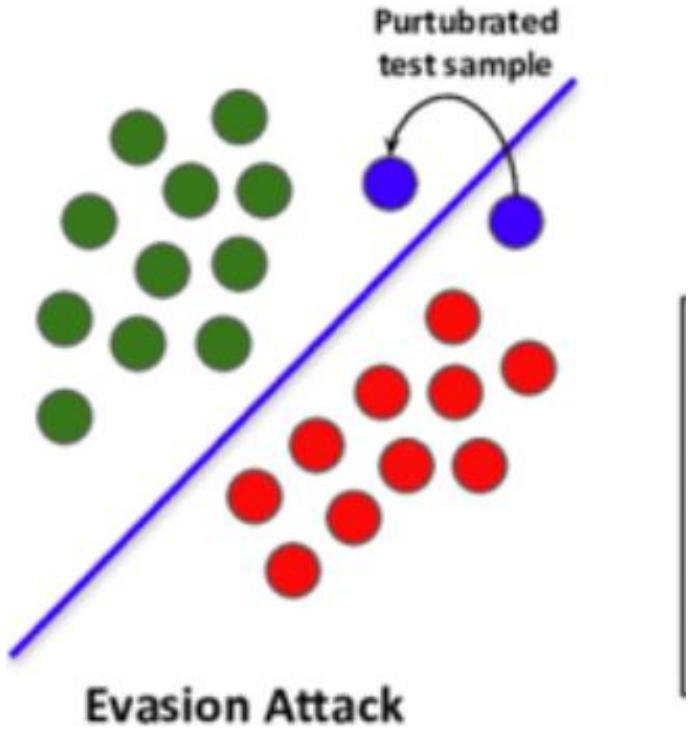
An arms race

- Two players, playing against each other
- Both adapt their strategy on the other person move
- Adversarial behaviour

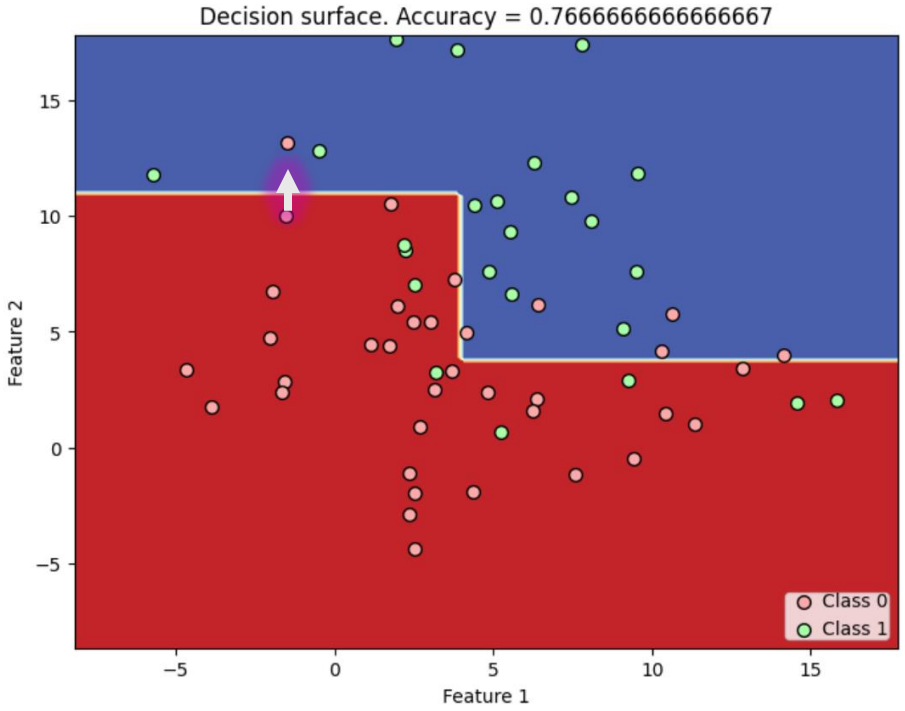


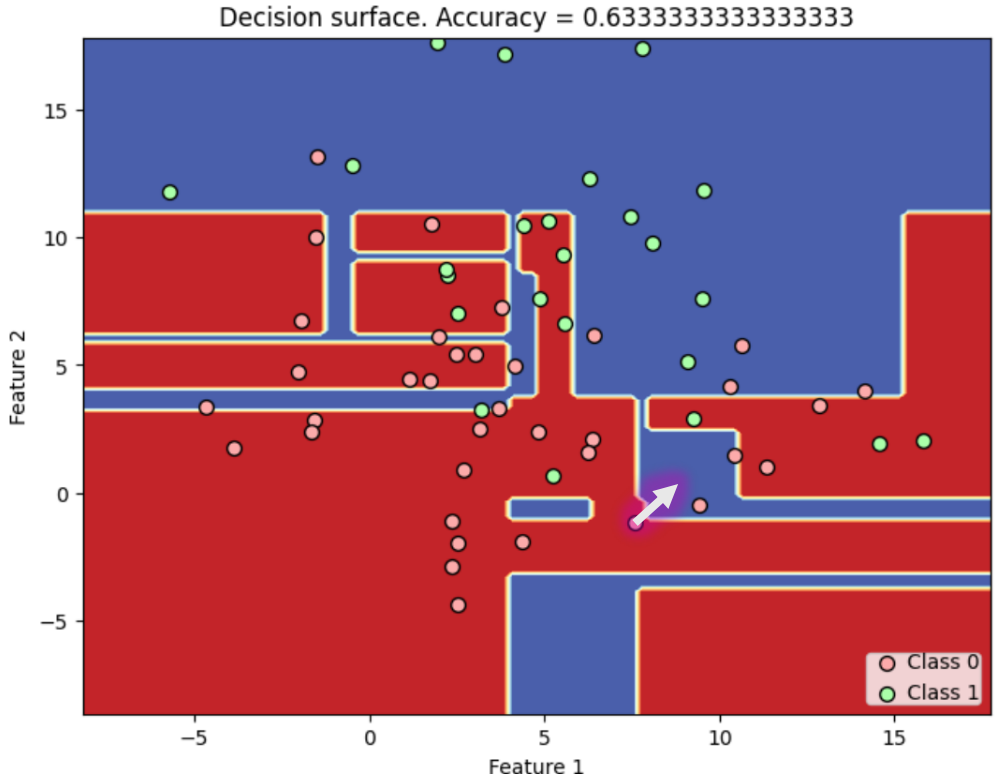
Attacking the boundaries

- What if the fraudsters know the classifier we use?
- They can slightly modify their fraud
- They pass the test!
- Adaptive attacks against classifiers are called adversarial attacks

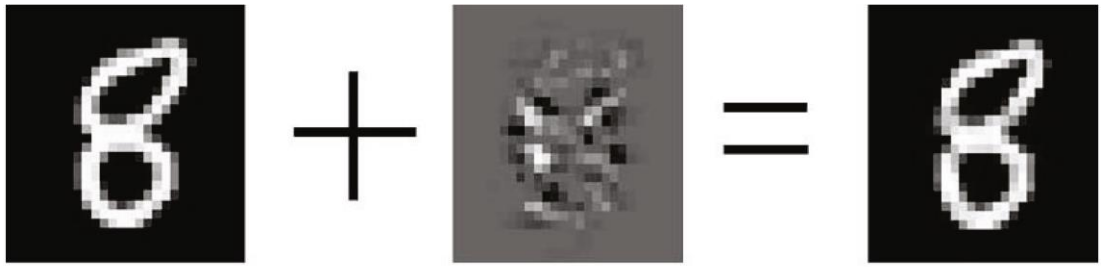


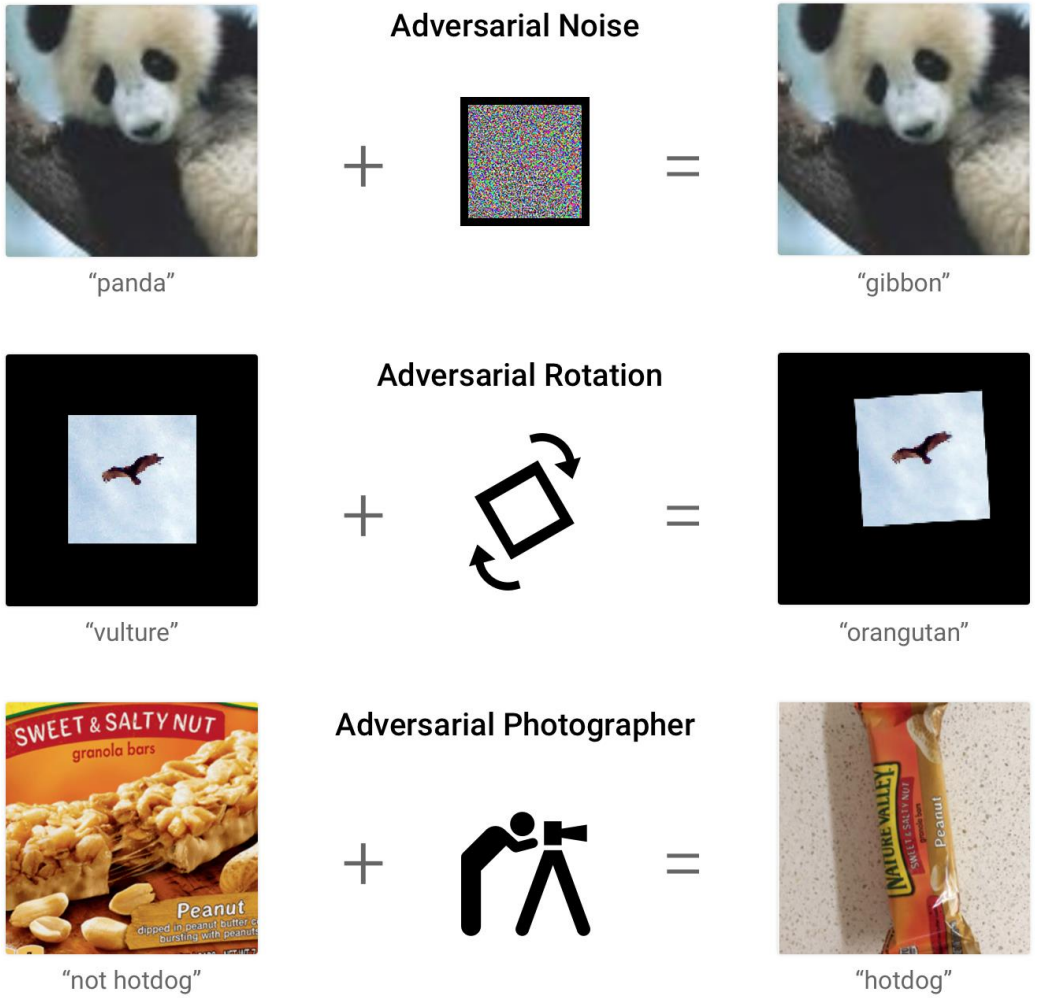
Application to the decision tree





Off Topic: Adversarial Machine Learning for images





My current work: reinforcement learning for AML

- Attackers learn from experience and craft increasingly effective attacks
- Some features may be unknown to the attacker, we need to consider that
- Reinforcement learning comes in handy!
- Attackers can learn the policy while attacking

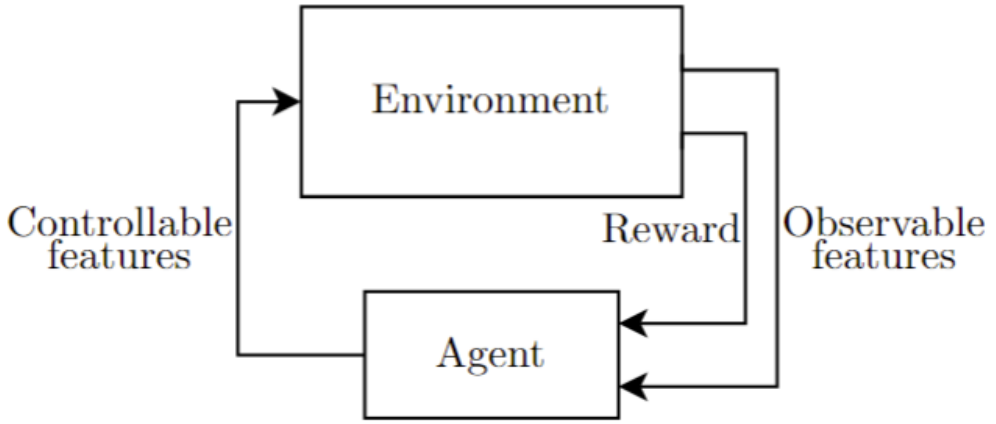


Figure 2: The environment

Conclusions

- Accurate models are good, we need to work on robustness
- Understand the threat posed by skillful attackers
- Improve the resistance of fraud detection engines

Thank you for your attention
Questions?

daniele.lunghi@ulb.be