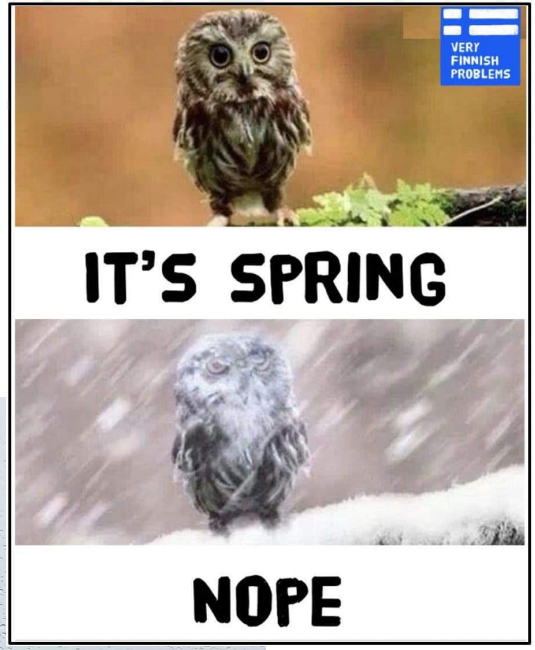
Machine Learning concepts

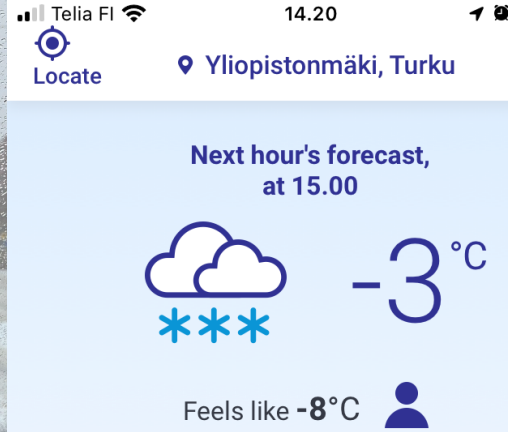
Luigia Petre lpetre@abo.fi
Åbo Akademi University, Finland
28.3.2023

Åbo Akademi University

Winter in Finland still



Åbo Akademi University



About me

- Associate professor in Computer Science @ Åbo Akademi University
  - Turku, Finland
- Born in Romania
- http://users.abo.fi/lpetre/

Plan

- What can Machine Learning (ML) do?
  - How well?
  - What are a few methods?
  - How does this help in gravitational waves search?
- This lecture is NOT
  - About python code
    - Although I point to notebooks
  - A directly applicable receipt for your own prediction/learning questions

What do we learn and how?

Machine Learning

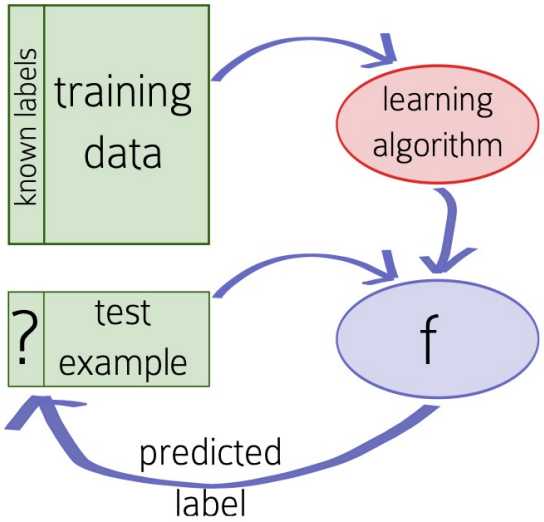
- We can’t say what makes people recognize a cat
- We can’t instruct the computer based on each pixel
- New type of programming paradigm
  - We let computers learn by example
  - Machine learning!
- Read more

What is learning?

- Assume Jane studied Algorithms
  - Read book, solved 100 exercises
  - She now takes exam
    - If exam is about French history
      - Not fair
    - If exam contains some of the 100 exercises
      - Kinda cheating if exam with open materials
- Generalization
  - Check that, given all she studied, when presented with a **new** exercise, she can apply what she learned, so that she is able to solve it

Supervised learning

- Collect some training examples
  - **Data** plus known **labels**
- Use the above to deduce a *learned function* `f`
- Use `f` to deduce **labels** of **new data**
  - Similar to training examples
  - Without labels
- **Training data** vs **test data**
  - Test data → great secret that we do not peek into
- Learning suceeds if
  - performance on test data is HIGH



Learning types (some)

1. *Regression*
   - predict a real value
   - value of a stock tomorrow given its past performance
   - predict Jane's score on machine learning final exam based on her homework scores
2. *Binary Classification*
   - predict simple yes/no response
   - predict whether Jane will enjoy a course or not
   - predict whether user review of newest Apple product is positive/negative
3. *Multiclass Classification*
   - put example into one of a number of classes
   - predict whether a news story is about entertainment, sports, politics, religion, etc
   - predict whether a CS course is Systems, Theory, AI or Other
4. *Ranking*
   - put a set of objects in order of relevance
   - predict what order to put webpages in, in response to a user query
   - predict Jane's ranked preferences over courses she hasn't taken

Why learning types?

- Measuring errors
  - We want “good” predictions
  - What does it mean for a prediction to be “good”?
- Regression
  - Predicting stock price that is *off* by €0.05 → much better than being off by €200.00
- Classification
  - Predicting “entertainment” *instead of* “sports” → no better/worse than predicting “politics”

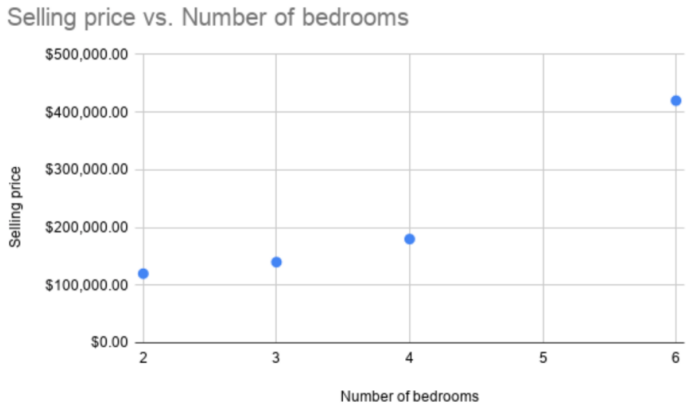
Regression

Linear Regression

- Machine Learning
  - How can I use X to predict Y ?
    - X → some information we have
    - Y → some information we want to know
- Classic example → price of house
  - Given house with 2 bedrooms (X=2)
  - How much can we sell it for? (Y=$?)
- Linear regression
  - Creates **linear** equation in which we input the given X and get a (**real number**) value for our **target** variable Y

Example

Number of bedrooms	Selling price
2	$120,000
3	$140,000
4	$180,000
6	$420,000



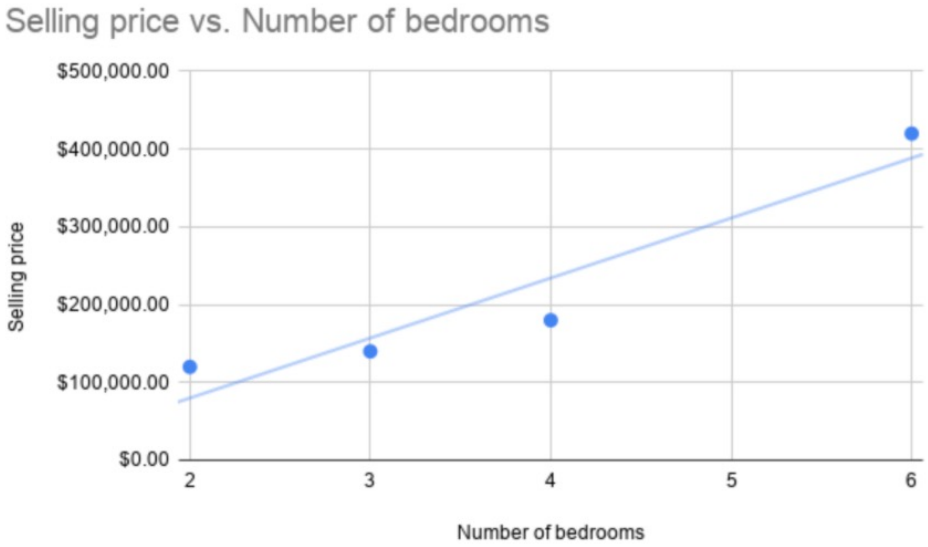
- Trend?
- Say we trained linear regression model to get equation of form:

```
Selling price = $77,143 * (Number of bedrooms) - $74,286
```

- Prediction

```
Your selling price = $77,143 * 2 bedrooms - $74,286 = $80,000
```

More than prediction



$$y = w_o + w_1 x$$

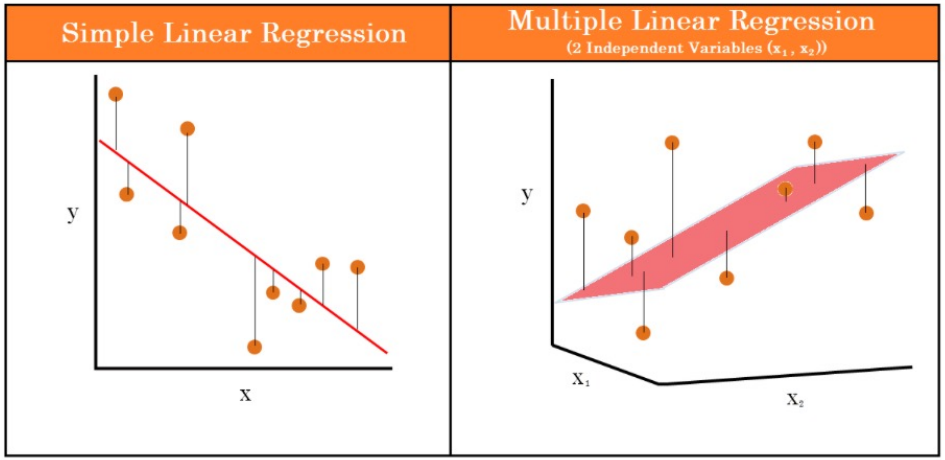
Simple regression

- Once trained, model takes form of linear regression equation
  - y → output variable
    - *Target variable* in ML, *dependent variable* in statistical modeling
    - Continuous value we are trying to predict
  - x → input variable
    - *Feature* in ML, *independent variable* in statistical modelling
    - Information given to us at any given time
  - w_0 → *Bias term* or y-axis intercept
  - w_1 → *Regression coefficient* or scale factor
  - w_i → *Weights*
- Goal → find values of unknown parameters of equation → values for weights w_0 and w_1

$$y = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

Multiple regression

- Regression → assume there is linear relationship between input variable(s) and output target variable
- Simple vs multiple regression
  - Main difference → number of independent variables that act as inputs



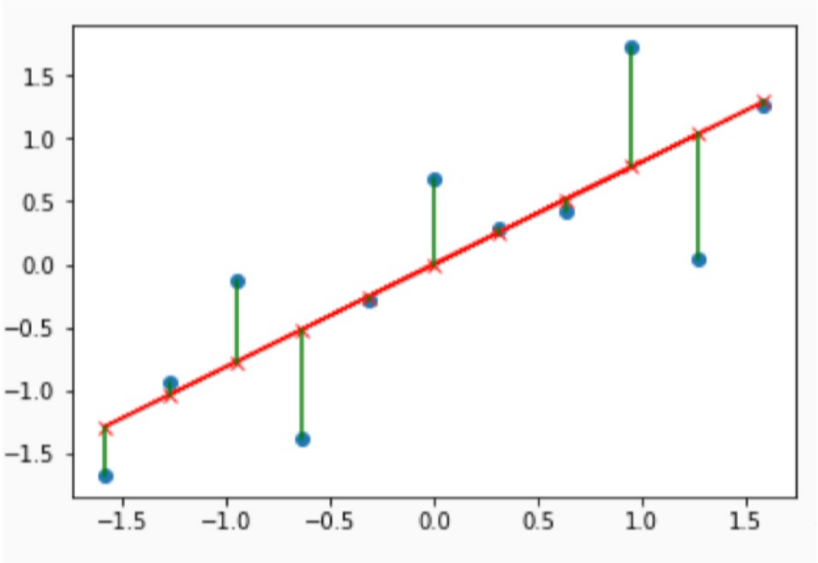
Training

- Goal → find weights w_i in linear equation $y = w_o + w_1 x$
- Steps (with *ordinary least squares*)
  1. Random weights initialisation
  2. Input initialized weights into linear equation + generate prediction for each observation point
  3. Calculate *Residual Squares Sum* (RSS)
     a) Calculate residuals
     b) Square residuals
     c) Add up residuals `1,600,000,000 + 293,882,449 + 2,946,969,796 + 987,719,184 = 5,828,571,429`
     d) Low RSS means we are doing well

Number of Bedrooms	Actual Selling Price	Predicted Selling Price	Residuals (Actual - predicted)	Residuals Squared
2	$120,000	$80,000	$40,000	$1,600,000,000
3	$140,000	$157,143	-$17,143	$293,882,449
4	$180,000	$234,286	-$54,286	$2,946,969,796
6	$420,000	$388,572	$31,428	$987,719,184

Number of bedrooms	Actual Selling price	Predicted Selling Price
2	$120,000	$80,000
3	$140,000	$157,143
4	$180,000	$234,286
6	$420,000	$388,572

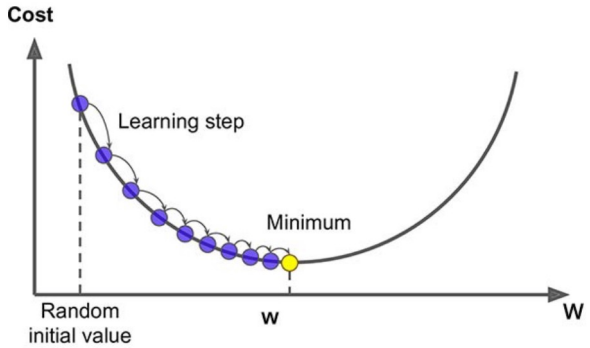
Doing well



4. Model parameter selection to minimize RSS
   - ML approaches → find best parameters for the linear model
   - Define **cost function**
   - Minimizing cost function
     - with **gradient descent**

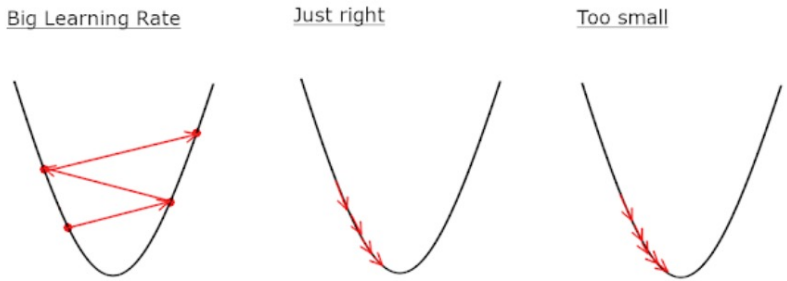
Gradient descent

- Method of changing weights based on RSS for each data point
  - Derivatives, etc, don't sweat it
    - Partial derivative of weight and bias → get slope of cost function at each point
  - Updates values for set of weights and bias
    - Based on slope
  - Re-iterates training loop over new values → moving a step closer to desired goal
  - Repeated until minimum error (cost function) reached
    - We cannot minimize cost function any further
    - We now have optimal weights
      - And can use model to predict



Devil in details…

- Hyperparameters…
  - Learning rate
    - how much are parameters changed at each iteration
    - Too high
      - model fails to converge and jumps from good to bad cost optimizations
    - Too low
      - model will take too long to converge to minimum error



Model evaluation

- Train data and test data!
- Metrics to check how good a fit the model is
  1. *Mean Squared Error* (MSE) → RSS divided by total number of observations/examples in our given dataset
     - average RSS is per data point
  2. *Root Mean Squared Error* (RMSE) → Square root over MSE
     - Much more intuitive for error interpretation
     - Equivalent to absolute error between our linear regression line and any hypothetical observation point
     - Can be directly used to interpret 'average error' that our prediction model makes

Model improvement

- Data pre-processing → cleaning data
  - Linear regression → several assumptions about structure of underlying data
  - If assumptions violated => model cannot make accurate predictions
  1. Remove outliers
     - Outliers in `y` → skew slope of the line disproportionately
     - Removed for better line fit
  2. Remove multicollinearity
     - Linear regression → assumes there is little/no correlation between the input values
       - If there is => it overfits data
     - Create correlation matrix for all features → check which pairs of features have high correlation
       - Remove these features to keep just one
  3. Assert normal distribution
     - Model assumes that independent variables follow a Gaussian distribution
     - Transform variables with `log`-transform (or smth else) if they are not normally distributed
  4. Assert linear assumption
     - If independent variables have no linear relationship with dependent variable, log transform them (..) to reshape polynomial relationships into linear

Model improvement - more

- Feature scaling
  - Numerical values of features can vary by different orders of magnitude
    - Eg: 1-10 bedrooms, 1000-10,000 square feet
  - Normalise and standardize features
- Regularization
  - Not useful for simple regression with one input variable
  - Commonly used in multiple regression to lower model complexity
  - Relates to coefficients/weights (of features) of a model
  - Can be thought of as **feature selection method**
    - Features with lower contributions to goodness of fit → removed and/or diminished in their effect (decrease their coefficient)
    - Important features → emphasized (increase their coefficient)

Why use linear regression?

1. Ease of use
   - Does not require lots of engineering overhead, neither before launch nor during maintenance
2. Interpretability
   - Unlike deep learning models (neural networks)
   - Linear regression → straightforward to interpret, no black box
3. Scalability
   - Algorithm → not computationally heavy
   - Scales well with increases in data volume (big data) and data velocity too
4. Performs well in online settings
   - Model can be retrained with each new example and generate predictions in near real-time
   - Not the case with computationally heavy approaches like neural networks or support vector machines

Regression types

1. Simple and multiple linear regression
2. Polynomial regression
3. Ridge regression and Lasso regression (upgrades to linear regression)
   - Special regularization techniques
4. Decision trees regression
5. Support Vector Machines (SVM)

References

- Some introductive words about ML
  - http://ciml.info/dl/v0_99/ciml-v0_99-ch01.pdf
  - From here : http://ciml.info (2017)
    - A course in Machine Learning, by Hal Daumé
- Keboola resources
  - https://www.keboola.com/blog/linear-regression-machine-learning
  - Notebook for diving deeper into the house example https://keboola.drift.click/b12a6d57-07be-4615-84e8-0779caf08c0c

To read more

- Classification vs regression vs prediction
  - https://towardsdatascience.com/classification-regression-and-prediction-whats-the-difference-5423d9efe4ec
- While we share Cassie Kozyrkov materials
  - **Chief decision scientist** at Google
  - https://kozyrkov.medium.com/everything-youve-ever-wanted-to-know-about-machine-learning-b396b0abee8c
    - Free youtube course on ML
    - Linear regression with smoothie example
      - Predict number of calories in smoothie based on some features
      - Feature engineering

Logistic regression

What is logistic regression?

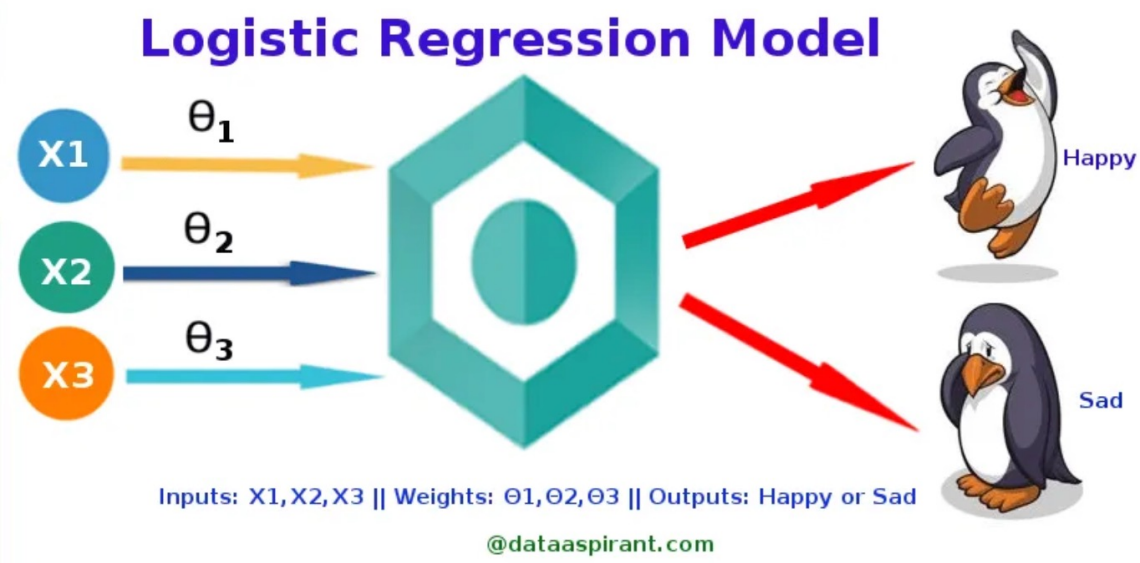
- Machine learning algorithm
  - Used to predict probability that observation $obs \in \{class_1, class_2, class_3, \dots\}$
    - Classification!

$$y = \frac{1}{1+e^{-(w_o + w_1 x)}}$$

- Regression or classification?

With penguins

$$y = \frac{1}{1+e^{-(w_o + w_1 x)}}$$



Who is who?

- y → predicted probability of belonging to default class
  - *Target variable* in ML (*dependent variable* in statistical modelling)
  - Categorical value that we are trying to predict
- Binary classification
  - default class → 1, the other class → 0
- $1/(1+e^{-z})$ → sigmoid function
- $w_0 + w_1 x$ → linear model within logistic regression

$$y = \frac{1}{1+e^{-(w_0 + w_1 x)}}$$

Example

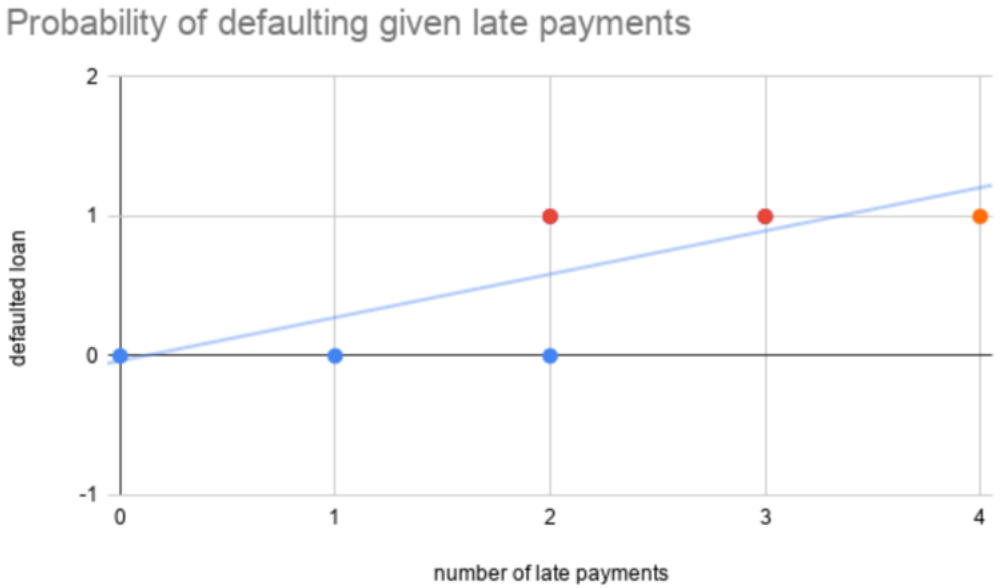
Client ID	Age	Gender	Loan size	Times payment was overdue	Defaulted?
1	32	F	$10,000	0	No
2	48	M	$30,000	12	Yes
3	51	F	$50,000	1	No

Client ID	Age	Gender	Loan size	Times payment was overdue	Defaulted?
1	32	F	$10,000	0	0
2	48	M	$30,000	12	1
3	51	F	$50,000	1	0

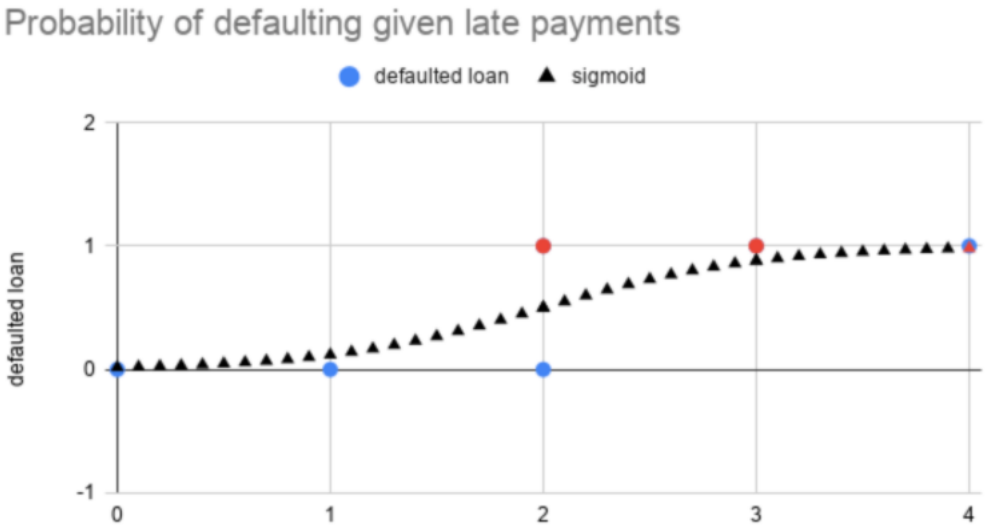
Client ID	Age	Gender	Loan size	Times payment was overdue	Predicted default
4	31	F	$12,000	1	0.13
5	28	F	$37,000	14	0.98
6	56	F	$48,000	2	0.23

Why not linear regression?

Number of late payments	Defaulted loan
0	0
1	0
2	0
3	1
2	1
2	1
3	1
4	1



Sigmoid function



Training

- Find out the best weights for our linear model within the logistic regression
- We compute optimal weights by optimizing the cost function
- Cost function
  - formal representation of objective that algorithm is trying to achieve

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log(h_\theta(x^{(i)})) + (1-y^{(i)})log(1-h_\theta(x^{(i)}))\right]$$

  - Principles behind it
    1. Checks average error between actual and predicted class membership
    2. Penalizes big errors + cases which are too confident (too close to 0 or 1)
  - Uses gradient descent → same as before, works with specific cost function

Model evalution

- Train data and test data!
- Metrics to check goodness of fit
  - **Accuracy** → percentage of correctly classified samples
    - 90% → logistic regression model correctly classified 90% of all examples
  - **ROC AUC** → Area Under the Receiver Operating Characteristic Curve
    - relationship between true positive rate (TRP) and false positive rate (FPR)
      - TRP → ratio of samples that we correctly predicted belonging to the correct class
      - FPR → ratio of samples for which we incorrectly predicted their class membership
    - preferable to accuracy
      - multiclass prediction settings or for class imbalance problem

Model improvement

- Logistic regression uses linear model
  - Suffers from similar issues
- Data preprocessing
  1. Remove outliers
  2. Remove multicollinearity
  3. Assert linear assumption
  4. Assert normal distribution
- Logistic regression → Additional assumptions => additional needs for cleaning
  1. Binary output variable → Transform output variable into 0 or 1
  2. Failure to converge
     - Assumption → no single variable will perfectly predict class membership
     - If feature perfectly predicts target class
       - Algorithm will try to assign it infinite weights (so important) => will fail to converge to a solution

- Feature scaling
- Regularization

Logistic regression types

1. Binary logistic regression
   - Target variable takes one of two possible categorical values
   - Eg: spam vs. not spam, 0 vs. 1, dog vs. not dog, etc
2. Multinomial logistic regression
   - Target variable takes one of three/more possible categorical values
   - Eg: vote Republican vs. vote Democratic vs. No vote, or “buy product A” vs. “try product A” vs. “not buy or try product A
   - ‘One vs. all’ method instead
     - We choose target class (let’s say A) and calculate the probability of A versus all of the other classes (B and C and…)
     - We repeat the method for each class
3. Ordinal logistic regression → similar to multiple logistic regression, except target categorical variables are ordered
   - Eg, medals at the Olympics

Classification

- Support Vector Machines
- Classification decision trees
- Random forest classification
- K-nearest neighbours
- Naive Bayes classifier

Resources

- A blog entry from Kaboola
  - https://www.keboola.com/blog/logistic-regression-machine-learning
- A notebook with the example
  - https://keboola.drift.click/c0a19919-134f-441c-8ef9-03934fde1934

Classification

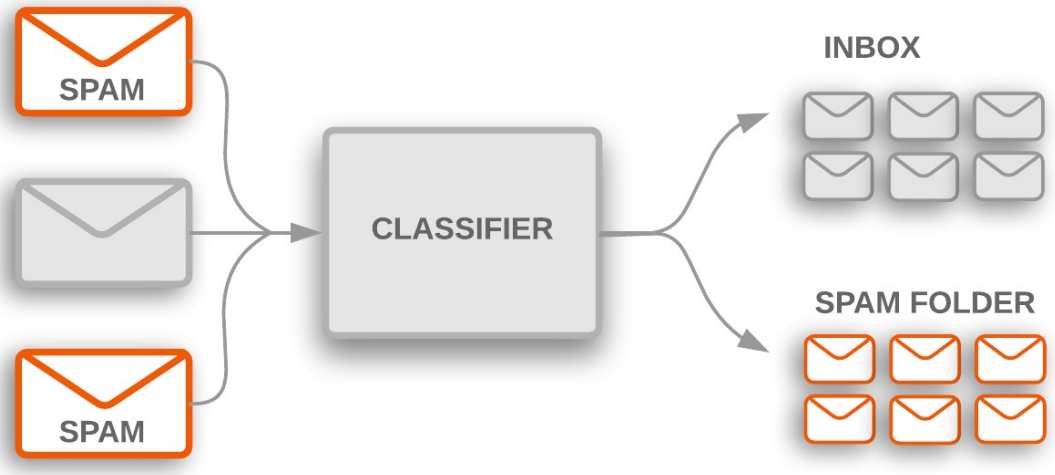
There is no classification?

- According to Cassie Kozyrkov
- Classification → what we call it when the desired output is categorical

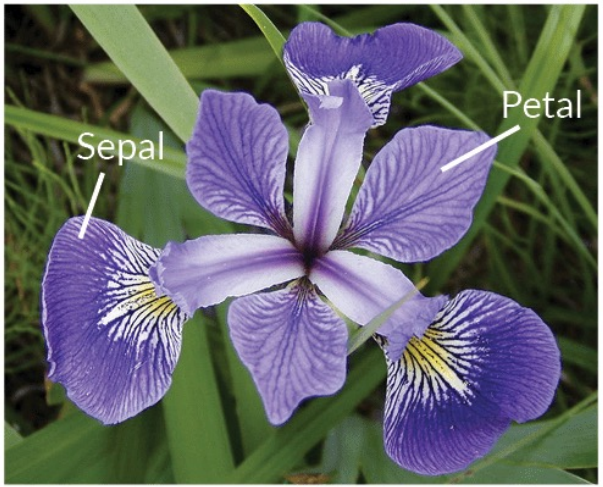
4 main types

1. Binary Classification
2. Multi-Class Classification
3. Multi-Label Classification
4. Imbalanced Classification

Binary classification



Multi-class classification



Iris Versicolor

Iris Setosa

Iris Virginica

Multilabel classification

- Special type of classification task
  - Multiple output variables for each instance from the dataset
  - One instance can have multiple labels

Imbalanced classification

- Instances of dataset → biased or skewed distribution
- One class of input variables has higher frequency than the others
- Eg → detecting fraudulent transactions in credit card transactions
  - Much fewer examples to train on

Classification algorithms

1. Naive Bayes Classifier
2. Logistic Regression
3. Decision Tree
4. Random Forests
5. Support Vector Machines
6. K-Nearest Neighbour
7. K-Means Clustering

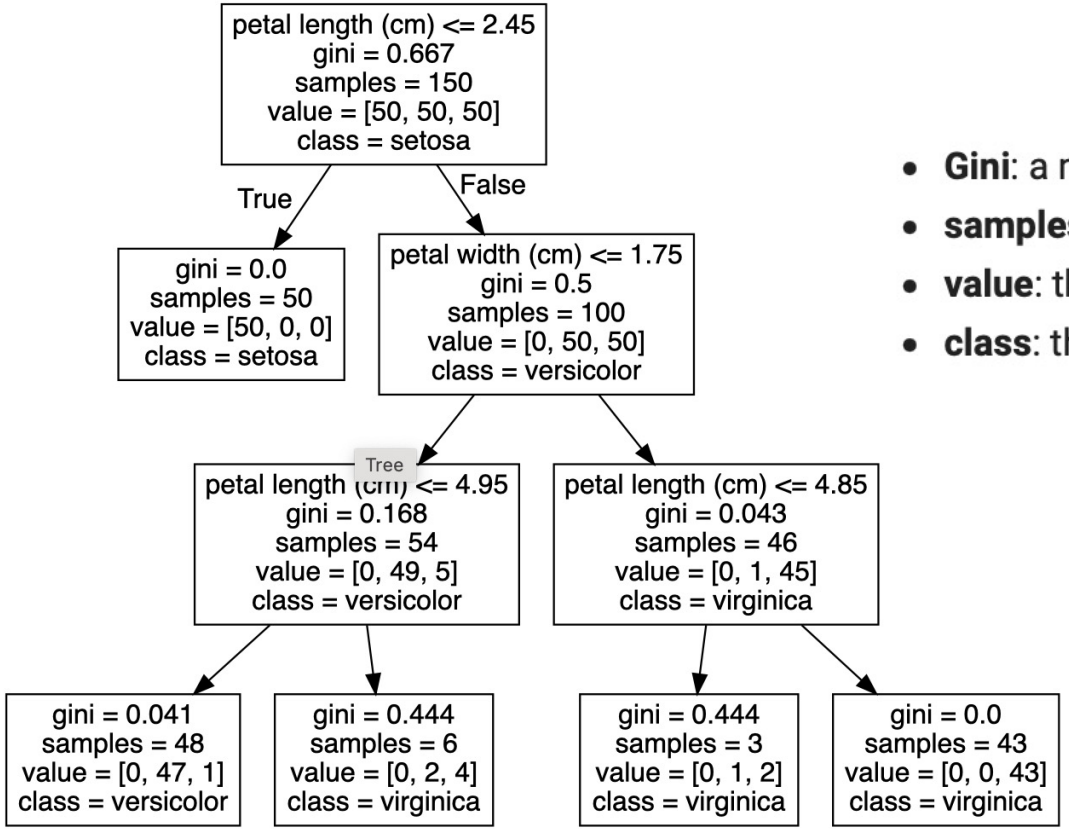
Decision trees

- Decision trees
  - general class of machine learning models, inherently NON-LINEAR technique
  - used for classification + regression
- Trained model
  - based on a series of questions with yes/no answers
  - resulting tree structure contains all combination of responses

Tree-based models

- Popular
  - mimic human decision making process
  - work well for a large class of problems
  - naturally handle multiclassification
  - handle a mix of categorical and numerical data
  - easy to understand and explain
- Disadvantages
  - They may create complex trees which sometimes become irrelevant
  - Not same level of prediction accuracy as compared to other algorithms

Iris tree



- **Gini**: a measure of node purity.
- **samples**: the number of observations in the node.
- **value**: the distribution of observations in each class.
- **class**: the most common label in the node.

How do we build a tree?

- Divide and conquer
  - what questions to ask
  - in what order
  - what answer to predict once you asked enough questions
- Greedy algorithm
  - computationally infeasible to consider all possible trees and choose best one
  - instead we build the decision tree greedily
    - **if I could ask only one question, what would I ask?**
    - we want feature which is most useful in advancing towards the goal

Gini

- Gini impurity metric
  - measure of node impurity
  - probability of misclassifying an observation if it were randomly labeled
    - based on the distribution of labels of the node
- Decision tree performs node splits that result in reducing the Gini
- For node m

$$G_m = \sum_k p_{mk}(1-p_{mk})$$

- p_{mk} → fraction of observations of class k in node m

Example

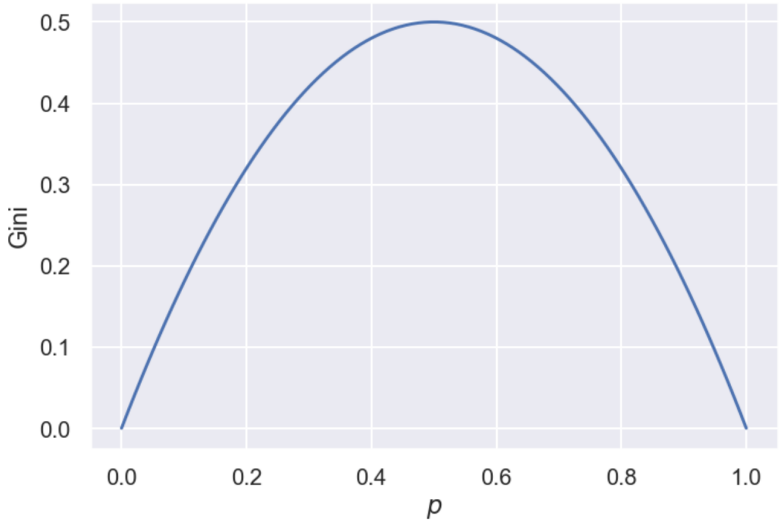
- 2 cases, 10 observations belonging to two classes
- Case 1 → each class has equal representation in the node [5, 5]
- Case 2 → only the first class is present [10, 0]
- Gini impurity

$$G_m = \sum_k p_{mk}(1-p_{mk})$$

$$G = \frac{5}{10}\left(1 - \frac{5}{10}\right) + \frac{5}{10}\left(1 - \frac{5}{10}\right) = 0.5$$

$$G = \frac{10}{10}\left(1 - \frac{10}{10}\right) + \frac{0}{10}\left(1 - \frac{0}{10}\right) = 0$$

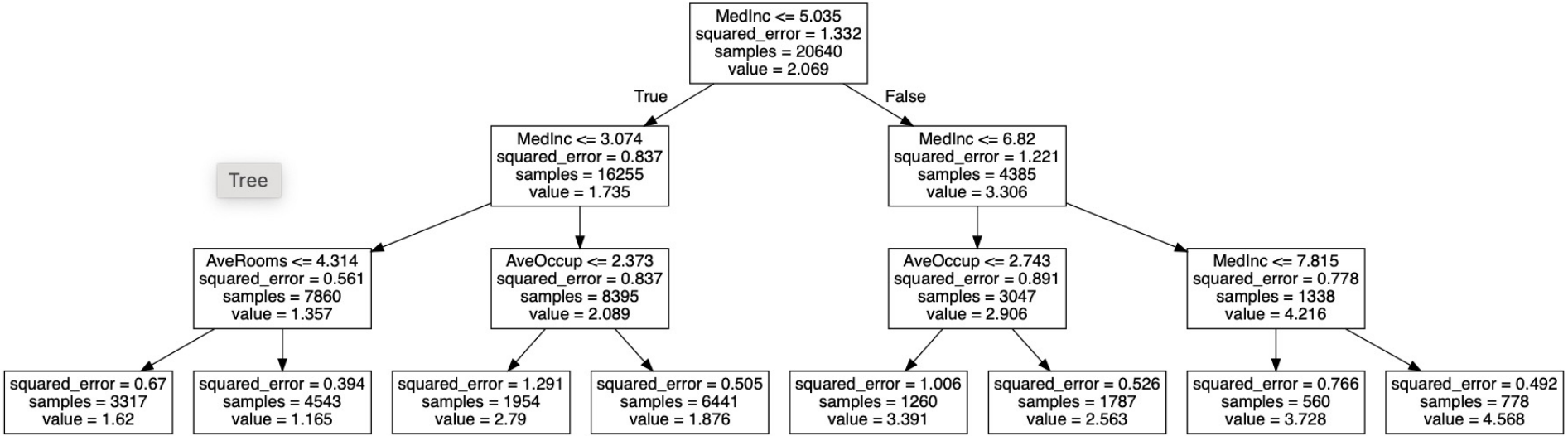
- The greater the node purity
  - the lower the Gini metric



Tree training

- Classification And Regression Tree (CART) algorithm
- Steps
  1. For each feature p, construct *thresholds* t_{pi}
     - Eg: threshold for feature petal width is “petal width ≤ 0.8 cm”
  2. Choose threshold resulting in greatest reduction of the *weighted* error metric
     - Eg: criterion “petal width ≤0.8 cm” → greatest drop in overall Gini impurity from resulting two new nodes
  3. Split the data set into two sets, nodes, using the chosen threshold
  4. Repeat the process on the *child* nodes until termination criterion is met

Regression decision tree



- **mse** → mean squared error using mean label value for the predictions
- **value** → mean label value of all observations in the node

Decision Tree Hyperparameters

- Max depth → key hyperparameter for decision trees
  - controls how deep the tree is allowed to grow => how adaptive the model is to fit the training data
- Depth gets greater => model gets more complex => higher likelihood for large error
- Other hyperparameters
  - (max) Number of features to consider when deciding the best split
  - (min) Number of samples required to consider split on an internal node
  - (min) Number of samples required for a leaf (terminal node)

Ensembles

- Aggregation of multiple models
- Final prediction → combination of predictions from components
- Component models can be trained
  - with different algorithms => approach problem from different perspectives
  - with the same algorithm
    - applied to different samples
    - using different tunings
- Typically not possible to interpret an ensemble model

Resources

- Project Pro article
  - https://www.projectpro.io/article/7-types-of-classification-algorithms-in-machine-learning/435
- Notebook with these and more
  - https://github.com/zerafachris/g2net_2nd_training_school_malta_mar_2020/blob/master/lectures/ML_Luigia_Petre/ML5_6_TreesAndEnsembles.ipynb

Some words..

Help! I want to use ML for searching/… for GW

- Feature space is very big
  - And unknown!
- Likely
  - Interpretable models will not work well → we miss more info than we know
    - Unknown unknowns
  - Deep learning models will have a better chance
    - Their way of learning not interpretable at the moment


Questions?