APS1070

Foundations of Data Analytics and Machine Learning

Winter 2022

Lecture 1:

- *Introduction*
- *Course Overview*
- *Machine Learning Overview*
- *K-nearest Neighbour Classifier*

Sinisa Colic and Samin Aref

Instruction Team

Instructor: Prof. Sinisa Colic

Instructor: Prof. Samin Aref

Head-TA: Ali Hadi Zadeh

TA: Haoyan (Max) Jiang

TA: Chris Lucasius

TA: Nisarg Patel

TA: Mustafa Ammous

Get to know the instruction team: https://q.utoronto.ca/courses/223861/pages/course-contacts

3

Communication

- Preferred contact method for a quick response: Piazza;
  1. Via Piazza Post to the “Entire Class”
  2. Via Piazza Question using Post to “Instructor(s)” - Type the specific person’s name from the list

  or type “instructors” to include us all

- Communication via email (APS1070 in subject line) is fine if you have a reason for not using Piazza for that question.

4

Emails

Instructor: Prof. Sinisa Colic	colic@mie.utoronto.ca
Instructor: Prof. Samin Aref	aref@mie.utoronto.ca
Head-TA: Ali Hadi Zadeh	a.hadizadeh@mail.utoronto.ca
TA: Chris Lucasius	christopher.lucasius@mail.utoronto.ca
TA: Mustafa Ammous	mustafa.ammous@mail.utoronto.ca
TA: Haoyan (Max) Jiang	haoyanhy.jiang@mail.utoronto.ca
TA: Nisarg Patel	npnisarg.patel@mail.utoronto.ca

- Please prefix email subject with ‘APS1070’

5

A little bit about your instructor …

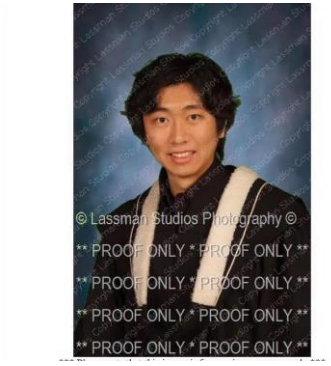
7

Instruction Team

Instructor: Prof. Sinisa Colic

Instructor: Prof. Samin Aref

Head-TA: Ali Hadi Zadeh



TA: Haoyan (Max) Jiang

TA: Chris Lucasius

TA: Nisarg Patel

TA: Mustafa Ammous

Get to know the instruction team: https://q.utoronto.ca/courses/223861/pages/course-contacts

Communication

- Preferred contact method for a quick response: Piazza;
  1. Via Piazza Post to the “Entire Class”
  2. Via Piazza Question using Post to “Instructor(s)” - Type the specific person’s name from the list

  or type “instructors” to include us all

- Communication via email (APS1070 in subject line) is fine if you have a reason for not using Piazza for that question.

Emails

Instructor: Prof. Sinisa Colic	colic@mie.utoronto.ca
Instructor: Prof. Samin Aref	aref@mie.utoronto.ca
Head-TA: Ali Hadi Zadeh	a.hadizadeh@mail.utoronto.ca
TA: Chris Lucasius	christopher.lucasius@mail.utoronto.ca
TA: Mustafa Ammous	mustafa.ammous@mail.utoronto.ca
TA: Haoyan (Max) Jiang	haoyanhy.jiang@mail.utoronto.ca
TA: Nisarg Patel	npnisarg.patel@mail.utoronto.ca

- Please prefix email subject with ‘APS1070’

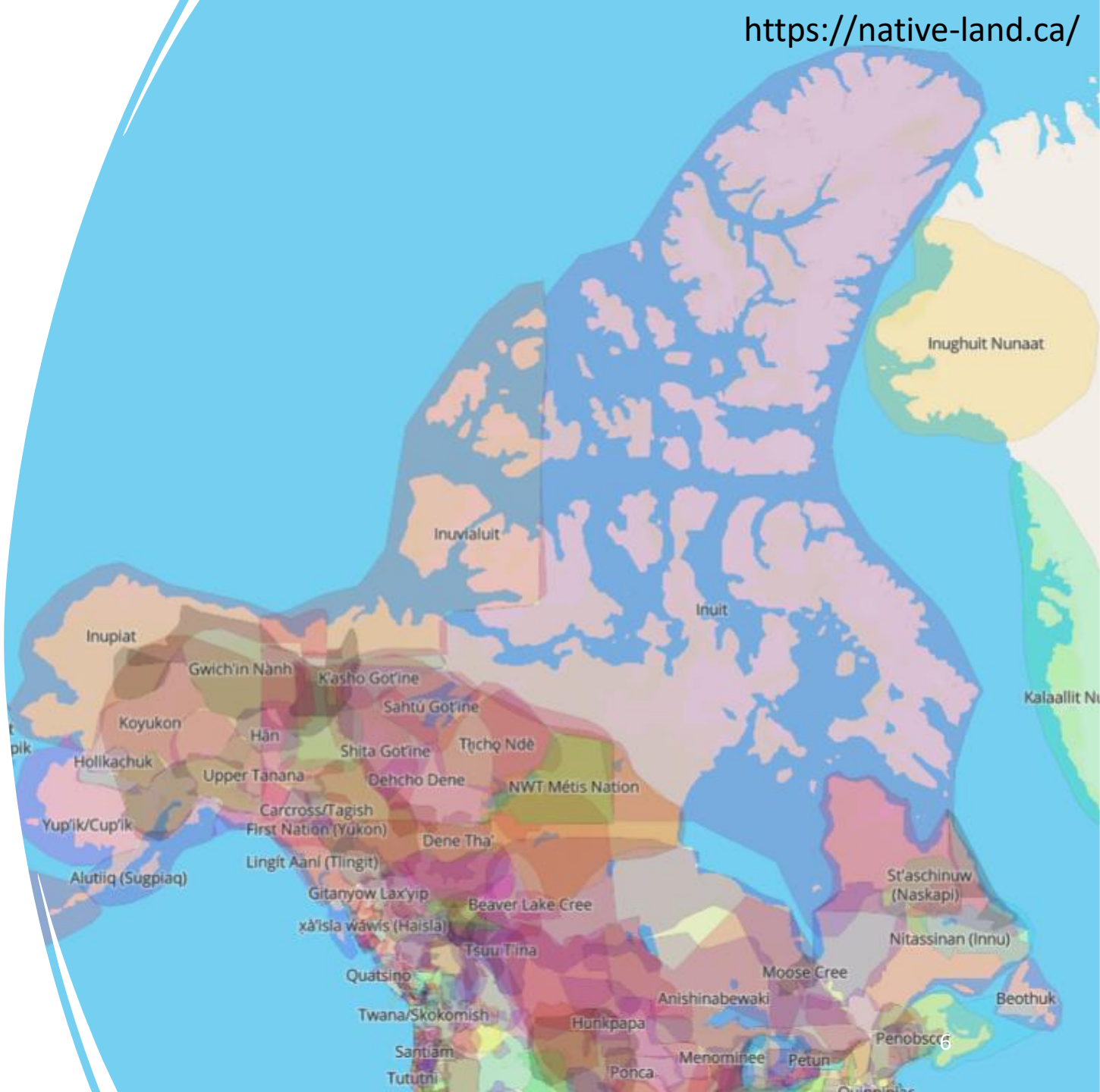
Traditional Land Acknowledgement

I wish to acknowledge the Indigenous Peoples of all the lands which we call home including the land on which the University of Toronto operates.

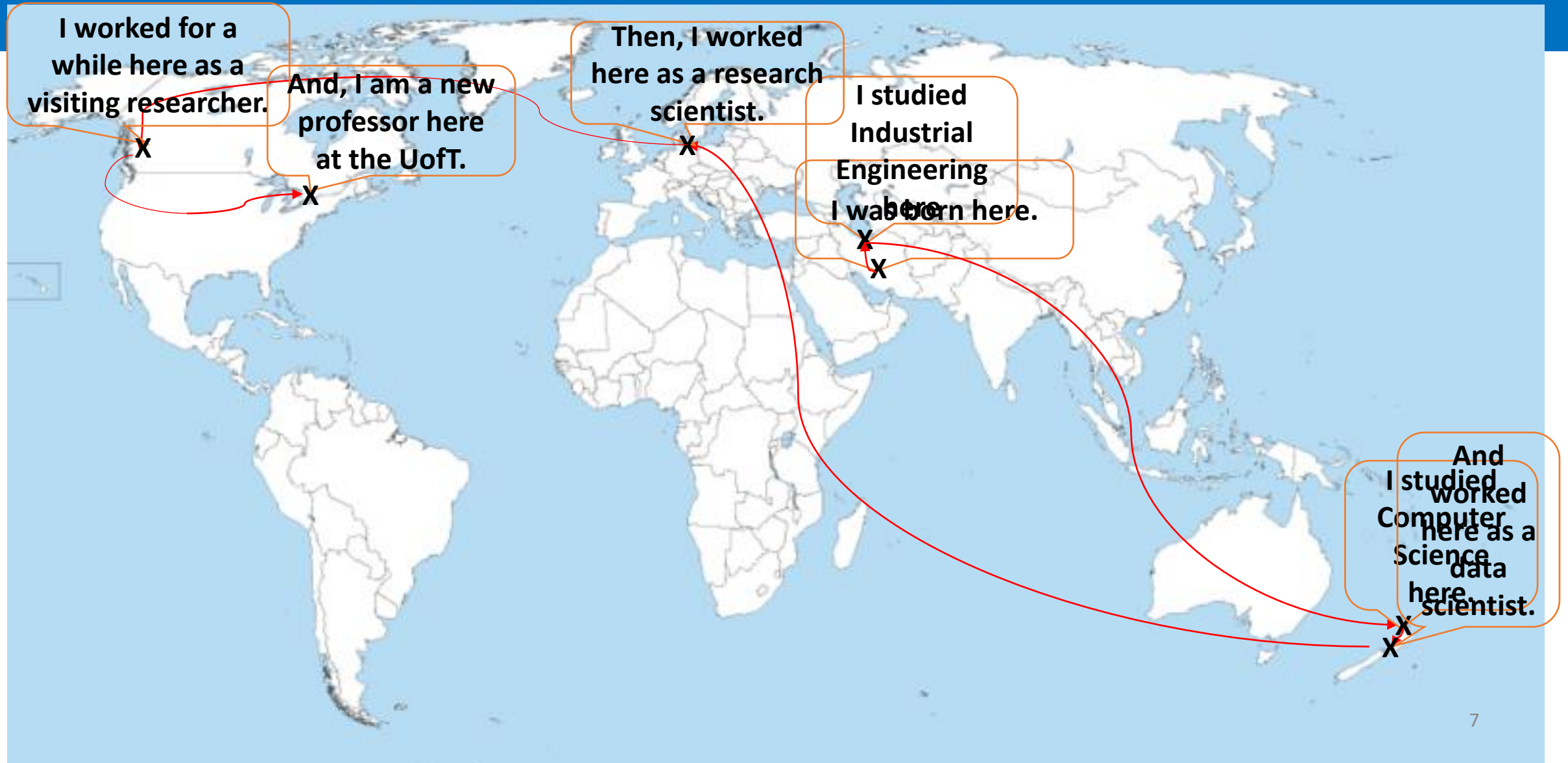
For thousands of years, it has been the traditional land of the Huron-Wendat, the Seneca, and the Mississaugas of the Credit.

Today, this meeting place is still the home to many Indigenous people from across Turtle Island and I am grateful to have the opportunity to work on this land.

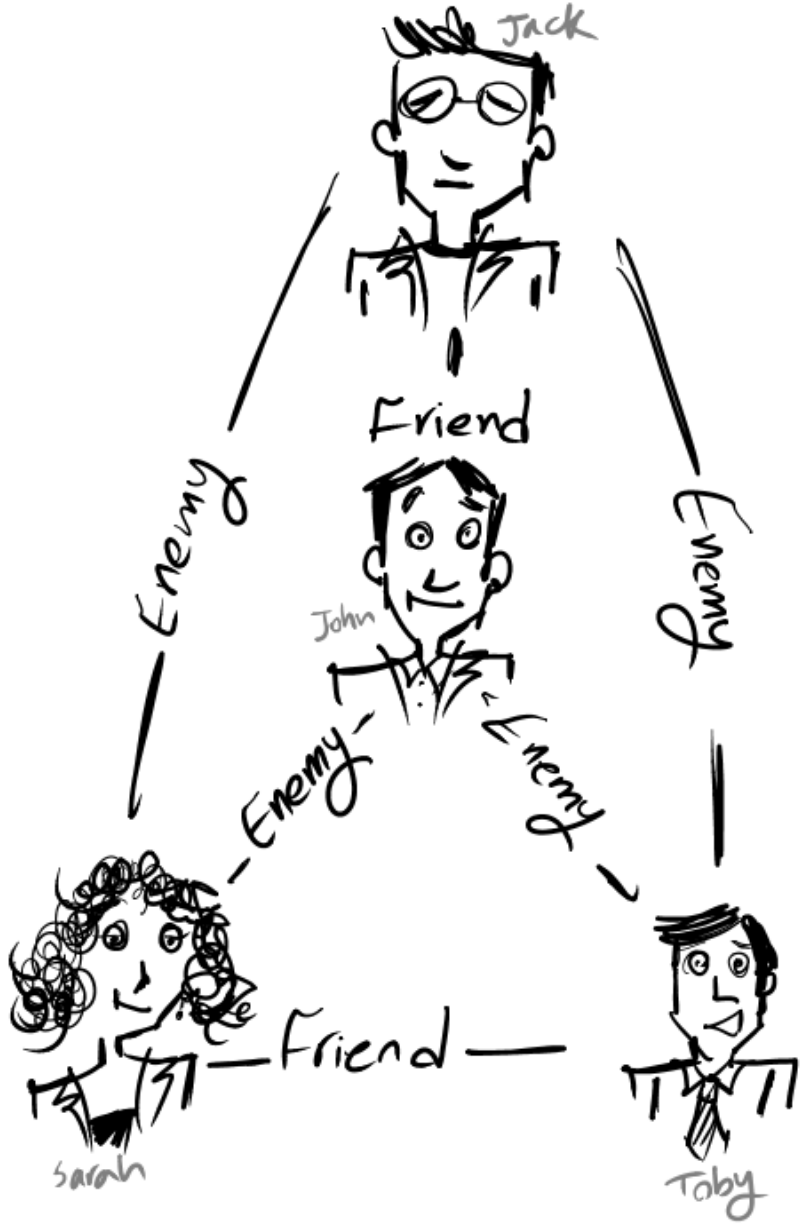
https://native-land.ca/



A little bit about your instructor …



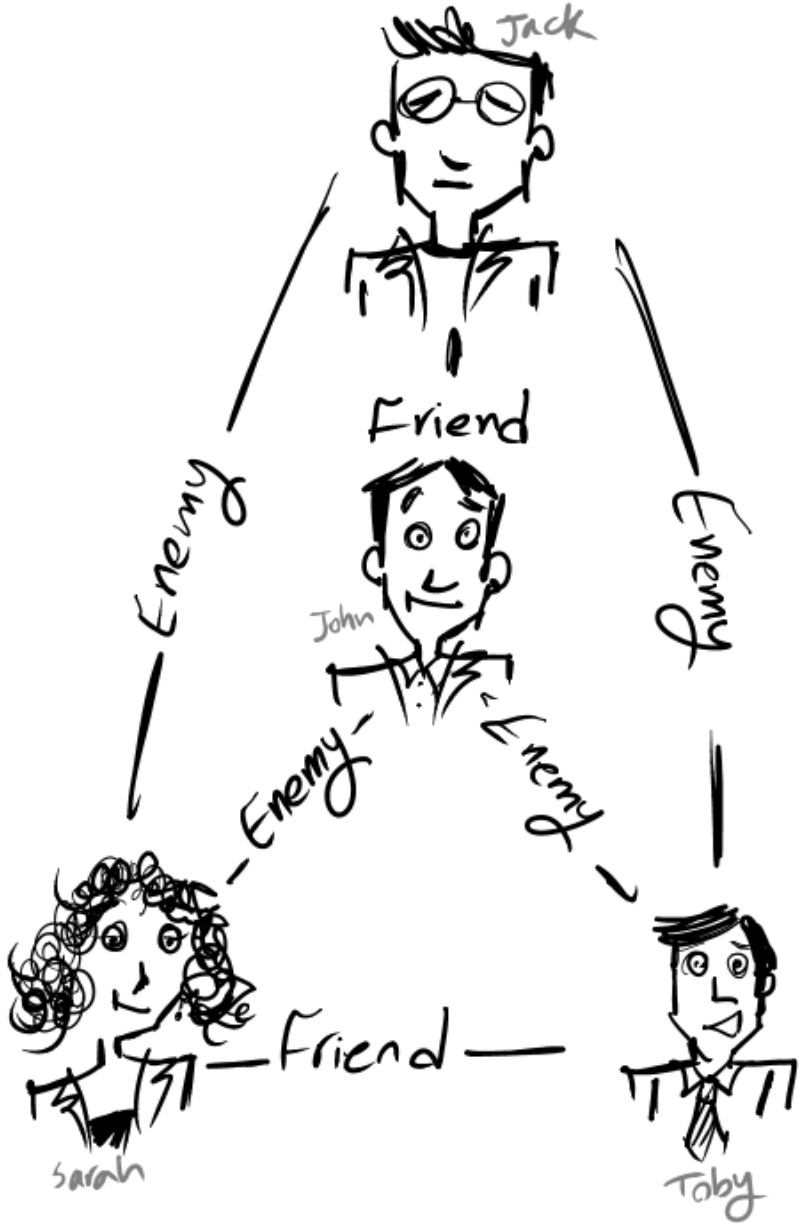

John
Jack
Sarah
Toby



Balanced network

In a *balanced* network:

Enemy of an enemy = friend
Enemy of a friend = enemy
Friend of an enemy = enemy
Friend of a friend = friend



Balanced network

In a *balanced* network:

Enemy of an enemy = friend ✓
Enemy of a friend = enemy ✓
Friend of an enemy = enemy ✓
Friend of a friend = friend ✓

Meet Mike, a friend of 

and 

.

Here is George, another friend of 

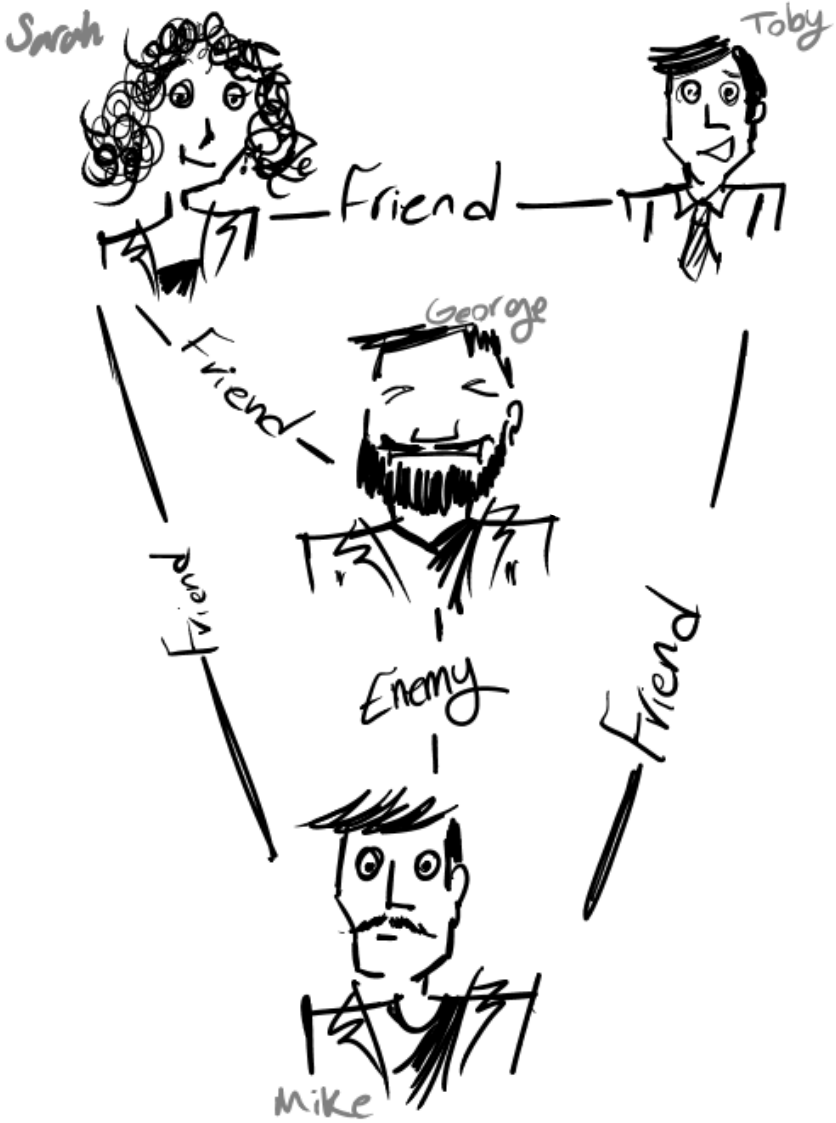
.

does not really know George that well.

knows George and they hate each other!

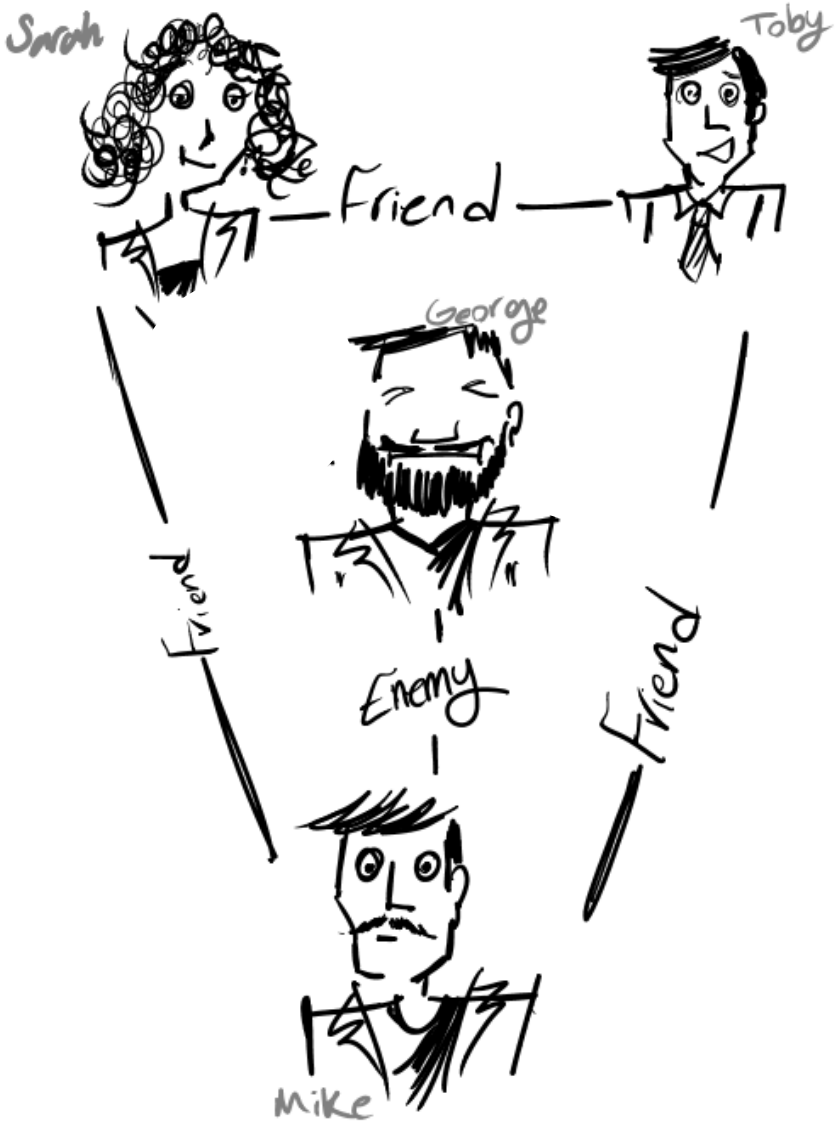
Unbalanced network

Enemy of an enemy = friend ✓
Enemy of a friend = enemy ✗
Friend of an enemy = enemy ✗
Friend of a friend = friend ✓



Balanced subgraph

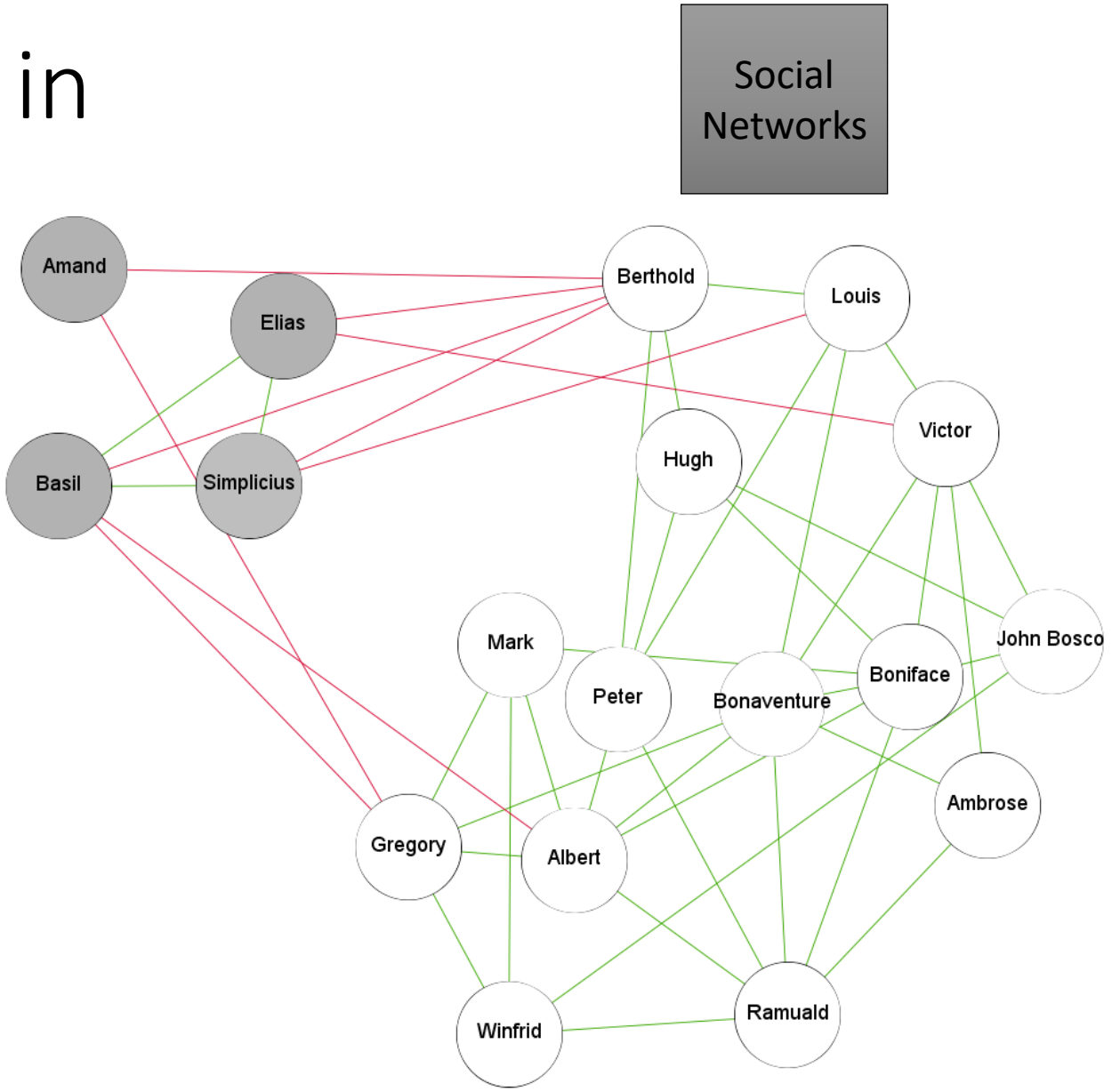
It is 1 edge away from balance.



Community detection in social networks

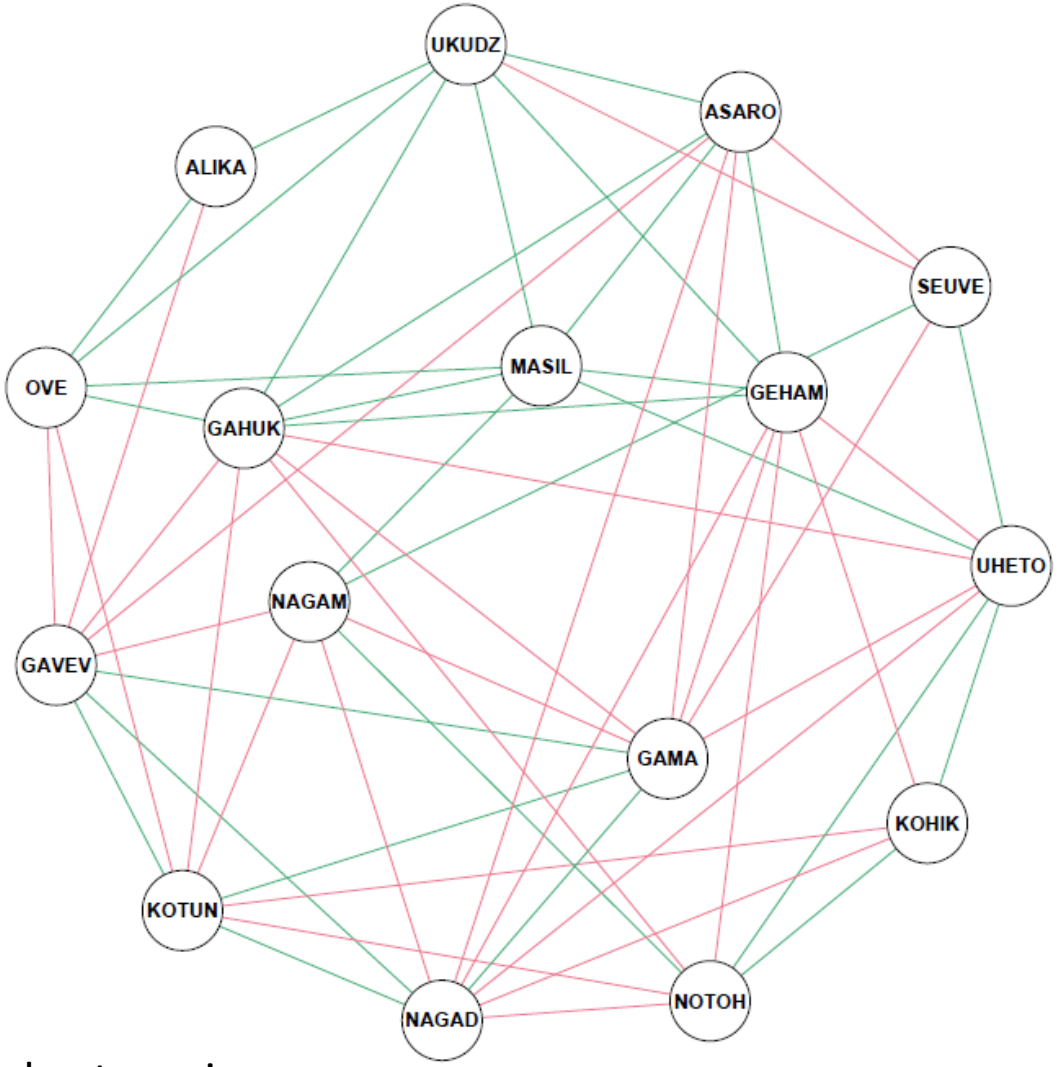
-positive relationship (green edge)
-negative relationship (red edge)

Nodes: people
Edges: positive or negative ties



New Guinean Tribes

Social Networks



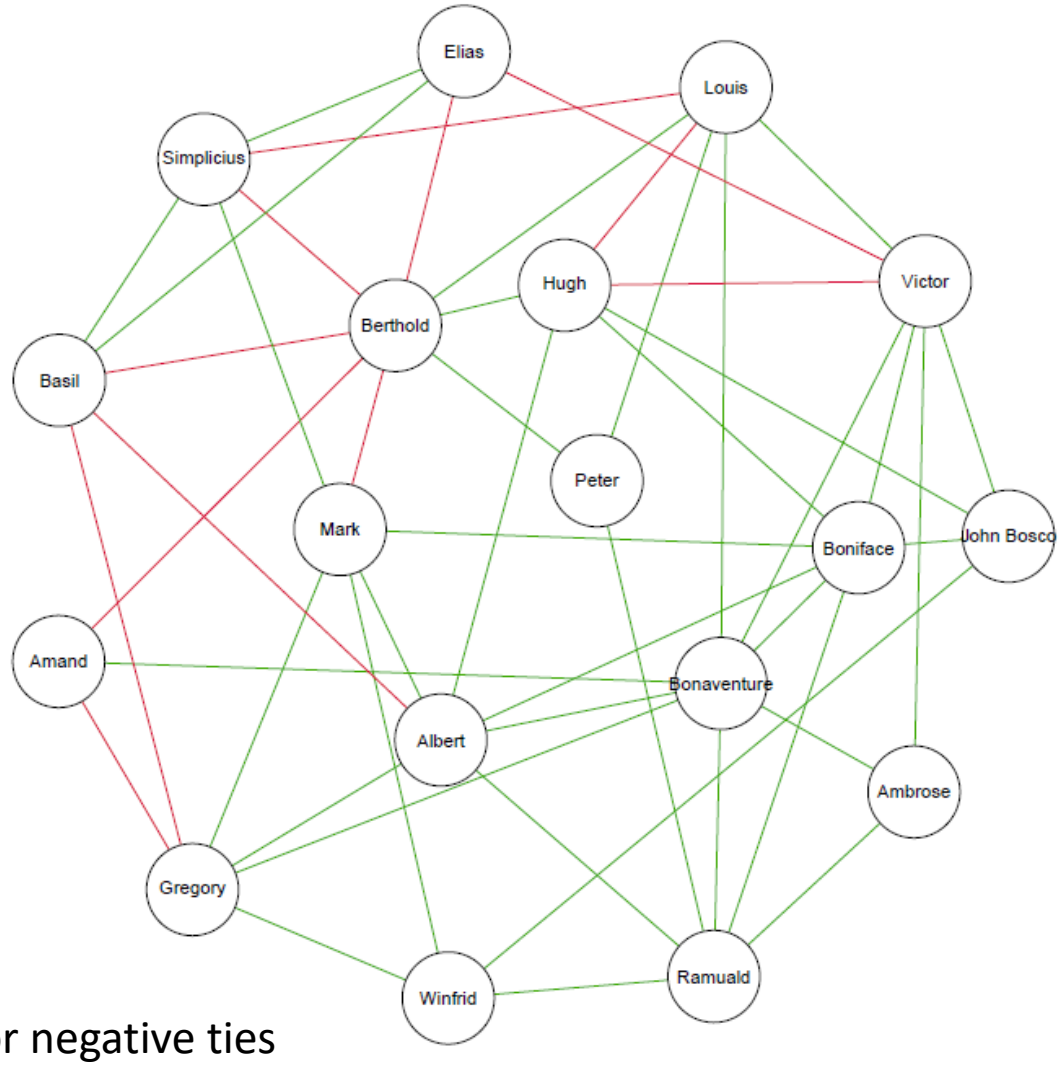
n	m	m^-
16	58	29

2^{16}= 65536 possible communities

Nodes: tribes
Edges: alliance and antagonism

Social Networks

Monks



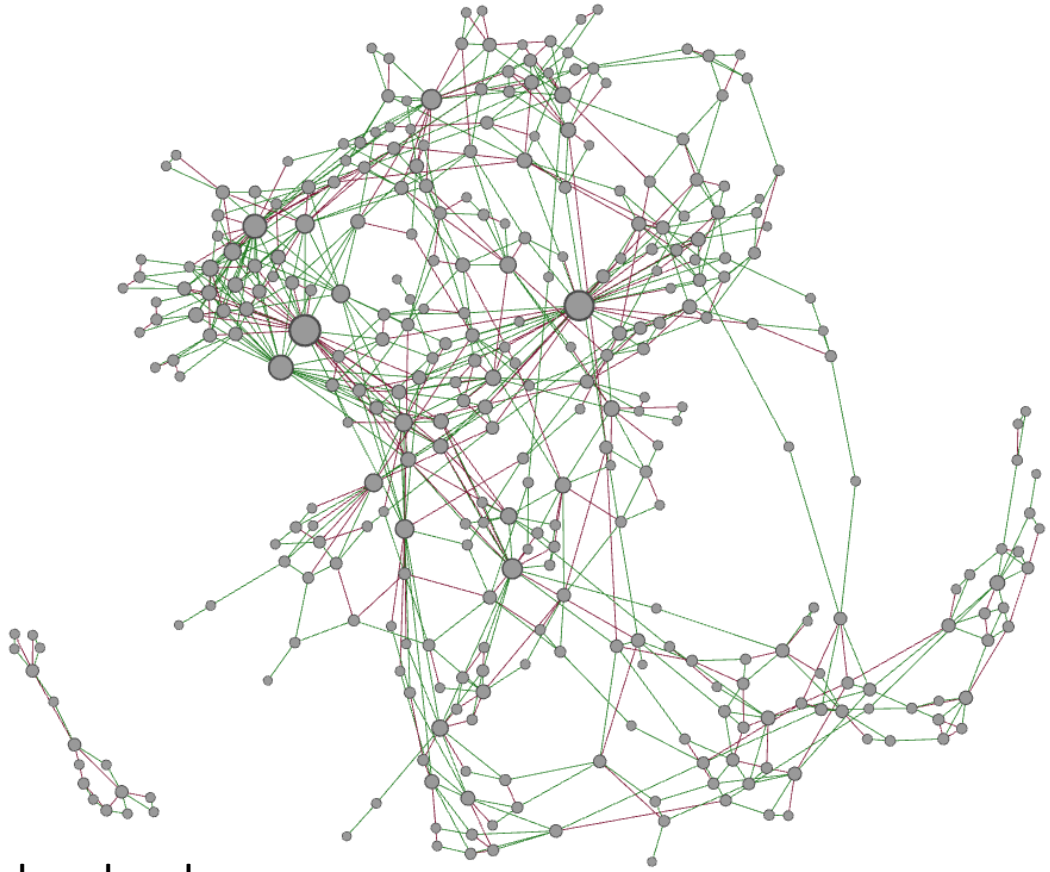
n	m	m^-
18	49	12

2^{18}= 262144 possible communities

Nodes: people
Edges: positive or negative ties

Biological Network of a Protein

Biology



n	m	m^-
329	779	264

2^{329}= 1 duotrigintillion !

Nodes: biological molecules
Edges: activation or inhibition relations

1979

Political Science

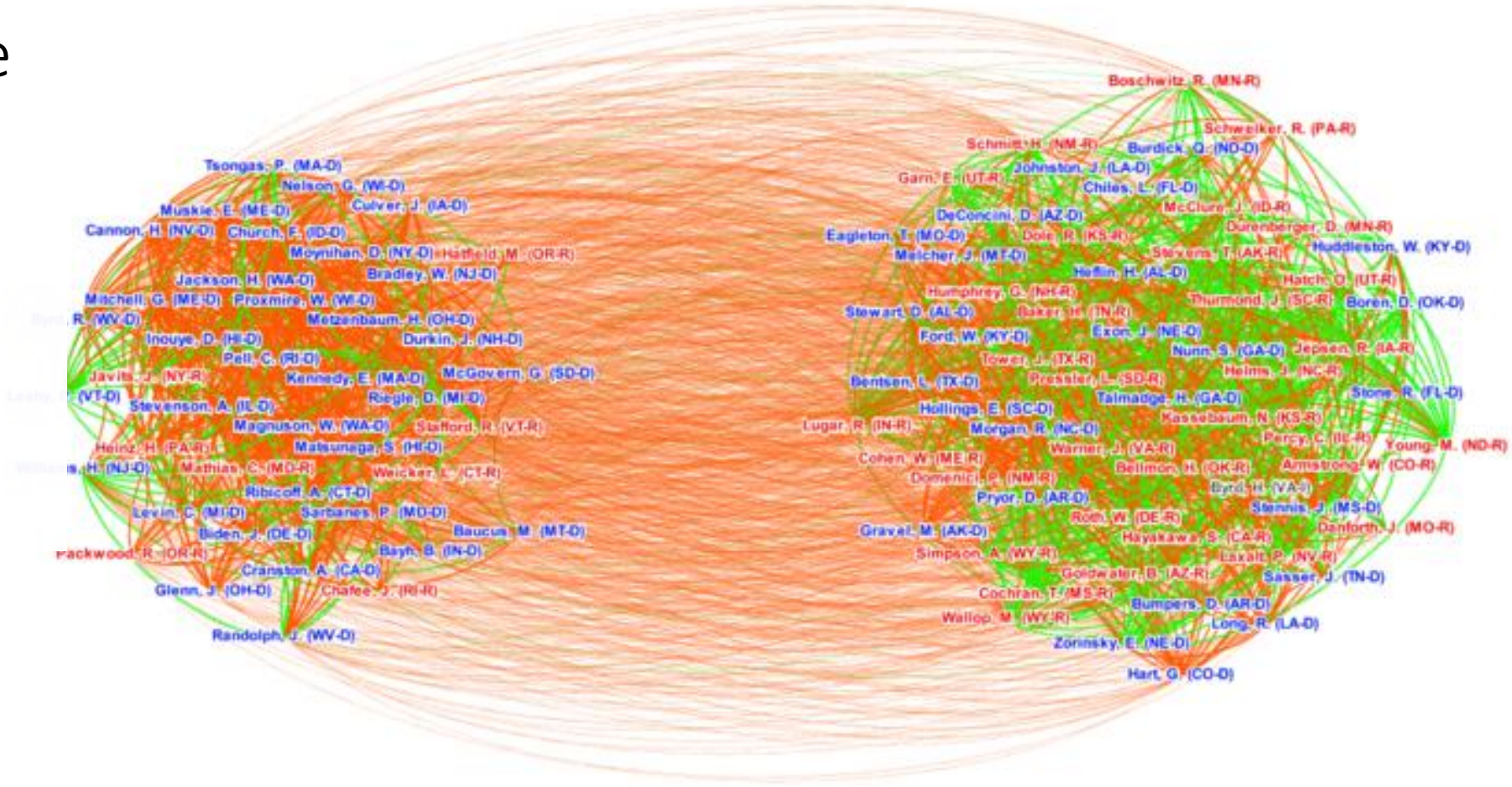
Partisan polarization in the US Senate over time

Node colors=party affiliation:

- Republican
- Democrat

Edge colors:

- Collaboration
- Avoidance



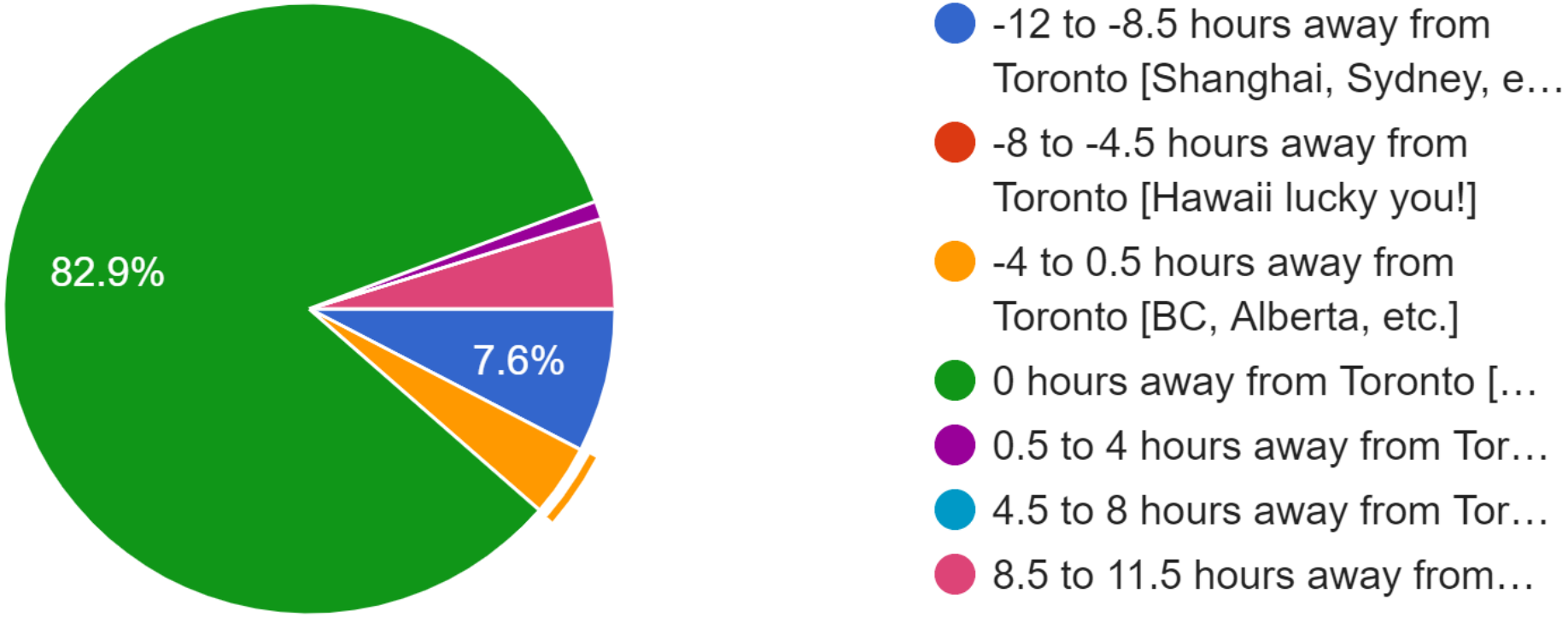
About you…

- What part of the world are you joining us from?
- What is your area of study?
- Previous experience?
- Why did you take this course?
- What do you want to get out of this course?

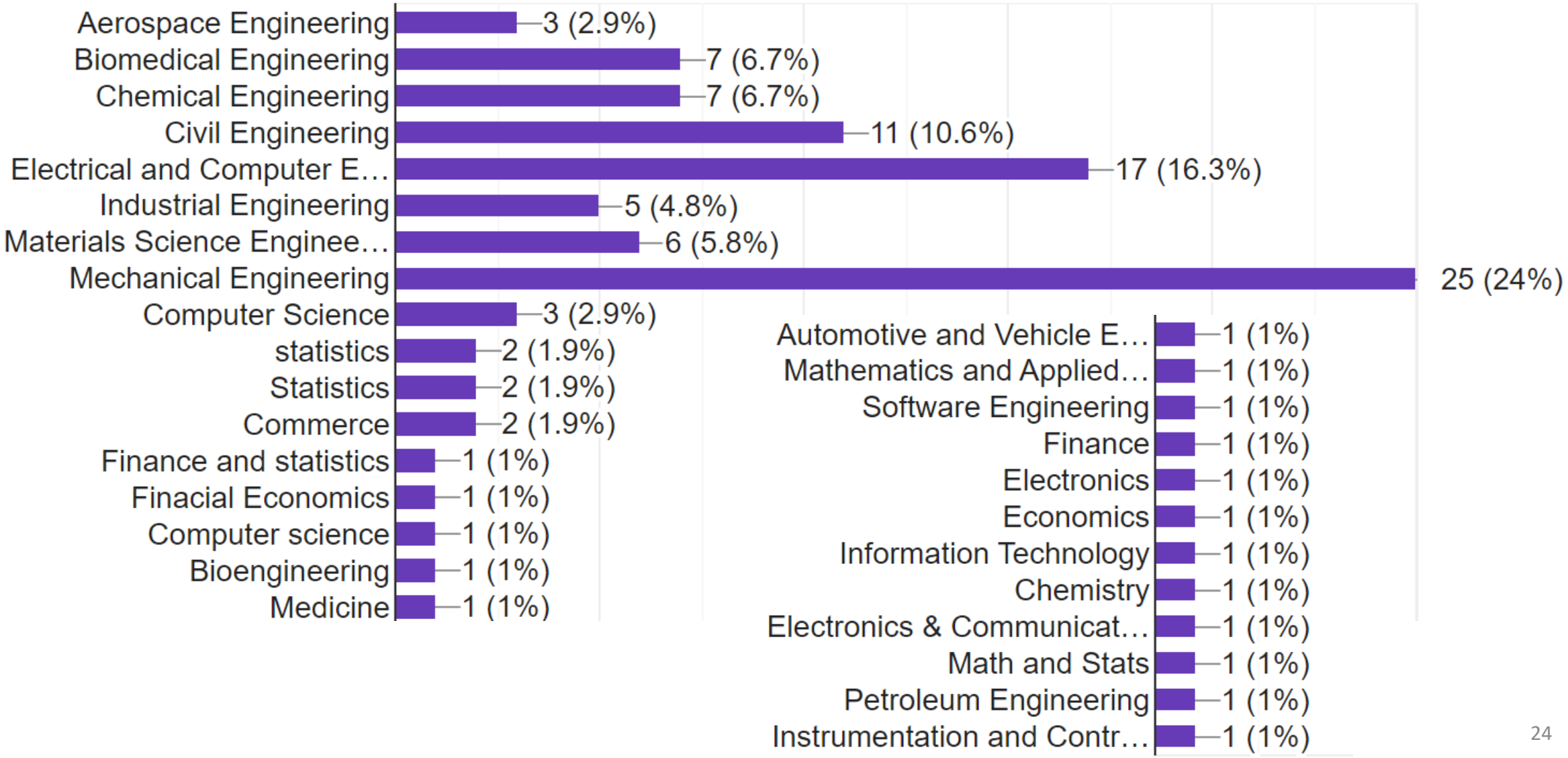
Survey: What is your time zone?

What is your time difference from Toronto?

105 responses



Survey: Undergraduate Degree?



Survey: Undergraduate Studies in Toronto?

Did you do your undergraduate studies in Toronto?

125 responses

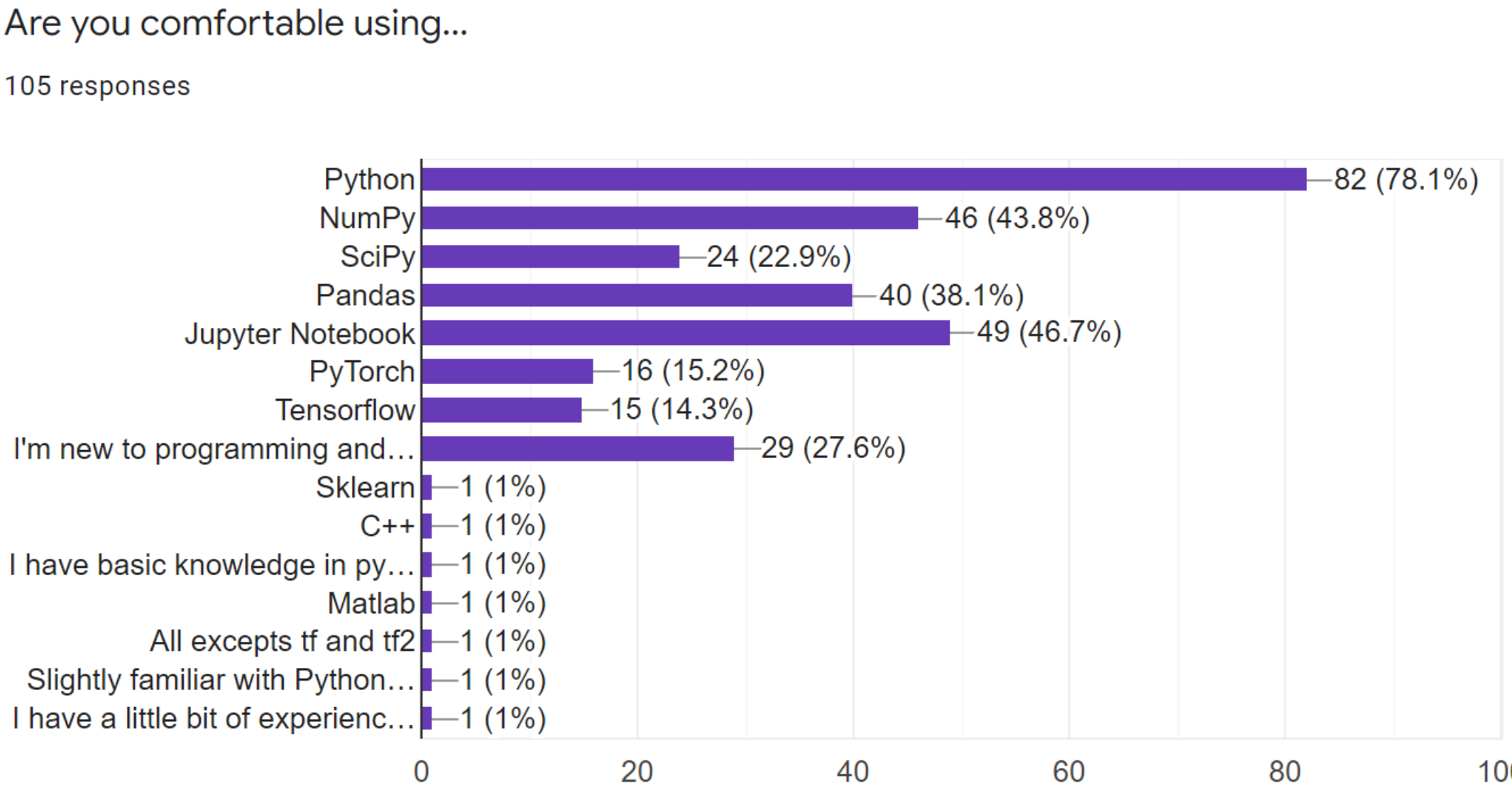
70.5% No

29.5% Yes

Survey: Why did you take this course?

- To become Machine Learning Researcher (~29.5%)
- To become a Data Scientist (~54.3%)
- Starting a Machine Learning Startup (~25.7%)
- For fun (~20%)
- …

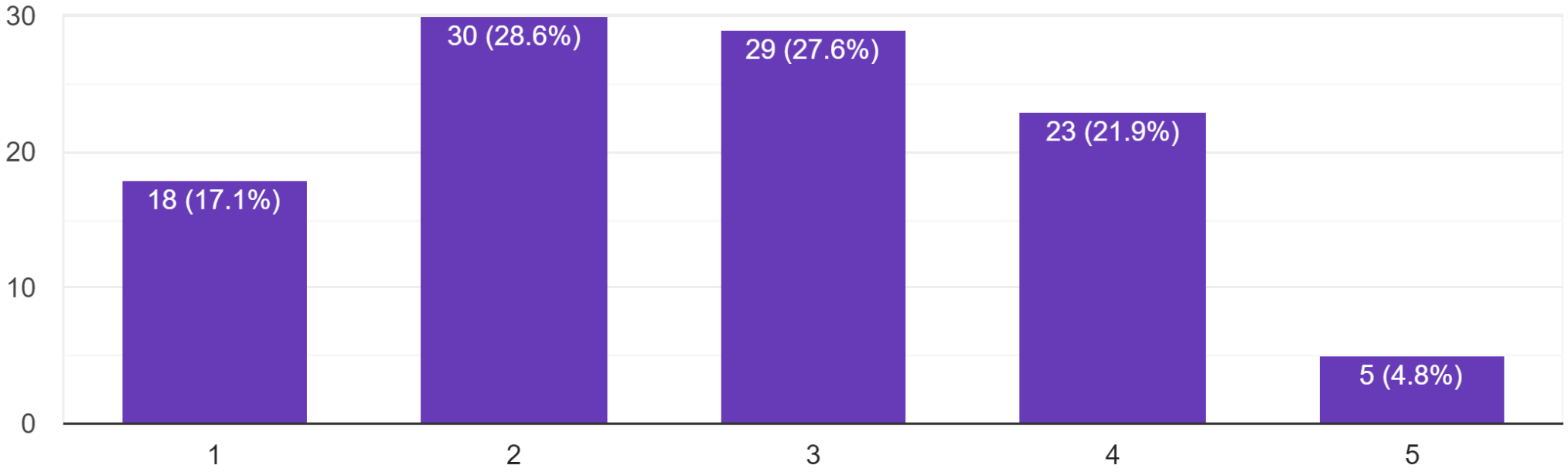
Survey: Programming Languages?



Survey: Rate Programming Abilities?

How would you rate your programming abilities? [1=Beginner, 5=Expert]

105 responses



Why are you taking this course?

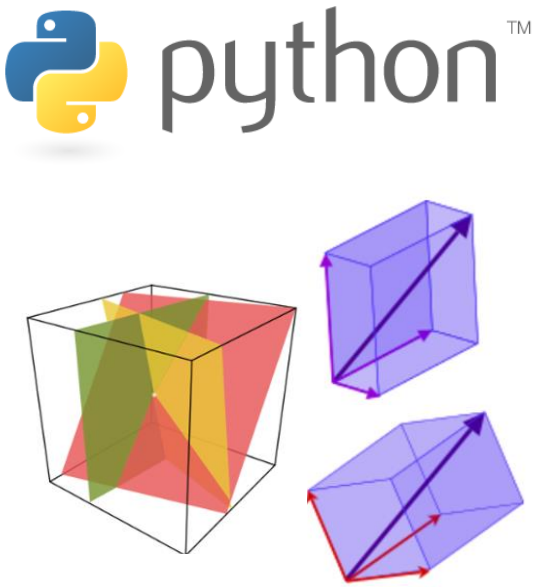
What kind of Projects would you like to do?

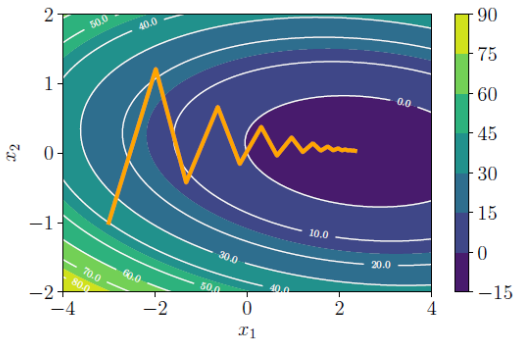
Part 1

Course Overview

Course Description

APS1070 is a prerequisite to the **core courses in the Emphasis in Analytics**. This course covers topics fundamental to data analytics and machine learning, including an **introduction to Python** and common packages, analysis of algorithms, **probability and statistics**, matrix representations and fundamental **linear algebra** operations, **basic algorithms** and data structures and **continuous optimization**. The course is structured with both weekly lectures and tutorials/Q&A sessions.





Primary Learning Outcomes

By the end of the course, students will be able to:

1. Describe and contrast machine learning models, concepts and performance metrics
2. Analyze the complexity of algorithms and common abstract data types.
3. Perform fundamental linear algebra operations, and recognize their role in machine learning algorithms
4. Apply machine learning models and statistical methods to datasets using Python and associated libraries.

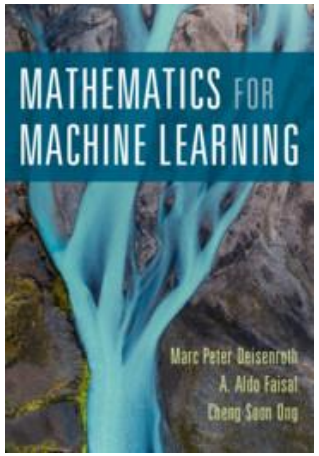
Course Components

- Lectures: Wednesdays (3 hrs)
- Tutorials/Q&A Sessions: Fridays (2 hrs)
- Four projects (submitted via **Quercus**)
- Eight tasks/quizzes for reading assignments (submitted via **Quercus**)

- Any material covered in **lectures / tutorials / projects / Piazza** is fair game for the midterm and final assessments.

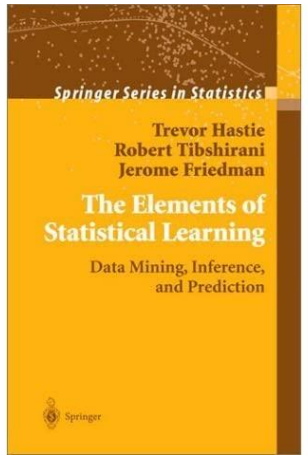
APS1070 Course Information

- Course Website: *http://q.utoronto.ca*
- Access course materials and Zoom Sessions
  - Verify using UTORid
- Textbooks:
  - **"Mathematics for Machine Learning" by Marc P. Deisenroth et al., 2020 (free)**
  - **"The Elements of Statistical Learning", 2nd Edition, by Trevor Hastie et al., 2009 (free)**
  - **"Introduction to Algorithms and Data Structures", 4th Ed, by Michael J. Dinneen, Georgy Gimel'farb, and Mark C. Wilson, 2016 (free)**
- Piazza discussion board for tutorial, project and almost all questions and communications.



Link

Link



Link

Computer Requirements and Online Tools

You will need access to:

- A computer **equipped with microphone and webcam (and ideally 2 screens)** to attend and participate in lectures and practical sessions on Zoom
- **Jupyter Notebook**, preferably through **Google Colab**, to be able to complete the projects
- **Quercus** for submitting projects and reading assignment tasks/quizzes and course announcements

Computer Requirements and Online Tools

You will need access to:

- **Piazza** to ask questions, communicate with the teaching team, and participate in course discussions
- Top 10 endorsed answerers (across the two sections of APS1070 for the Winter 2022) on Piazza with at least 3 endorsed answers get 2 points added to their final course grade.
- Questions in the general forms of “is this the correct answer?” or “what is wrong with my code?” or “why my code does not compile?” and the like will not receive a response.

Grade Breakdown

Projects/Quizzes	Weight (%)	Due dates
Reading assignment tasks/quizzes	4%	Deadline for reading assignment one: 17 Jan at 21:00 Deadlines for other reading assignments: As per course schedule
Project 1	10%	Due Feb 4 at 23:00
Midterm Assessment	14%	Tentatively scheduled for Feb 15 at 9:00 to Feb 16 at 15:00 (limited 2-hour window to start the exam and submit it)
Project 2	14%	Due Feb 28 at 23:00
Project 3	14%	Due Mar 11 at 23:00
Project 4	14%	Due Apr 1 at 23:00
Final Assessment*	30%	Tentatively Apr 12 at 9:00 to Apr 13 at 15:00
Bonus for Piazza+	(+2 points)	Apr 11

Penalty for late Submissions

Quercus submission time will be used. Late projects and reading assignments will incur a penalty as follows:

- -20% (of project maximum mark) if submitted within 72 hours past the deadline.
- A mark of zero will be given if the submission is 72 hours late or more.

Late midterm and exam submissions get a grade of 0.

Class Representatives

If you have any complaints or suggestions about this course, please email me directly. Alternatively, talk to one of the class reps who will then talk to us and the teaching team.

We need 2 class reps. Volunteers can send me an email (with “APS1070 class rep” in subject line) by 14 January.

Class reps are asked to keep in touch with the instruction team about any feedback they receive from students and to attend two staff-student meetings over the course of this semester.

This can be a great opportunity to develop your leadership skills.

Academic Integrity

- All the work you submit must be your own and no part of your submitted work should be prepared by someone else. Plagiarism or any other form of cheating in examinations, tests, assignments, or projects, is subject to serious academic penalty (e.g., **suspension or expulsion from the faculty or university**).
- A person who supplies an assignment or project to be copied will be penalized in the same way as the one who makes the copy.
- Several plagiarism detection tools will be used to assist in the evaluation of the originality of the submitted work for both text and code. They are quite sophisticated and difficult to defeat.

Software

- Python 3
- NumPy, Matplotlib, Pandas and many more
- Google Colaboratory
  - Jupyter notebook in the cloud
  - no installation required
  - requires Google Drive

All project handouts will be Jupyter notebooks

Why Python for Data Analysis?

- Very popular interpreted programming language
- Write scripts (use an interpreter)
- Large and active scientific computing and data analysis community
- Powerful data science libraries – NumPy, pandas, matplotlib, scikit-learn, dask
- Open-source, active community
- Encourages logical and clear code
- Invented by Guido Van Rossum →

Course Philosophy

- Top-down approach
  - Learn by doing
  - Explain the motivations first
  - Mathematical details second
- Focus on implementation skills
- Connect concepts using the theme of End-to-End Machine Learning

My goal is to have everyone leave the course with a strong understanding of the mathematical and programming fundamentals necessary for future courses.

Tentative Schedule (Weeks 1 – 3)

	Section	Date		Time	Topics
Week 1	*101 (Colic)*	Tuesday	11-Jan	12:00-15:00	**Introduction**
	102 (Aref)	Wednesday	12-Jan	9:00-12:00	Course Overview, K-Nearest Neighbours, Machine Learning Overview
	101 (Colic)	Thursday	13-Jan	13:00-15:00	Tutorial 0 - Python Basics and GitHub
	102 (Aref)	Friday	14-Jan	9:00-11:00	
Week 2	*Both*	*Reading assignment 1 Due*		*17-Jan*	*at 21:00*
	101 (Colic)	Tuesday	18-Jan	12:00-15:00	**Algorithms and Data Structures**
	102 (Aref)	Wednesday	19-Jan	9:00-12:00	Analysis of Algorithms, Asymptotic Notation, Sorting, Dictionary ADT, Hashing
	101 (Colic)	Thursday	20-Jan	13:00-15:00	Tutorial 1 - Basic Data Science
	102 (Aref)	Friday	21-Jan	9:00-11:00	
Week 3	*Both*	*Reading assignment 2 Due*		*24-Jan*	*at 21:00*
	101 (Colic)	Tuesday	25-Jan	12:00-15:00	**Data Exploration, Making Predictions, Foundations of Learning**
	102 (Aref)	Wednesday	26-Jan	9:00-12:00	End-to-End Machine Learning, Data Wrangling, Visualization, Decisions Trees
	101 (Colic)	Thursday	27-Jan	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	28-Jan	9:00-11:00	

Tentative Schedule (Weeks 4-6 and reading week)

Week 4	*Both*	*Reading assignment 3 Due*		*31-Jan*	*at 21:00*
	101 (Colic)	Tuesday	01-Feb	12:00-15:00	**Measuring Uncertainty and Evaluating Performance**
	102 (Aref)	Wednesday	02-Feb	9:00-12:00	K-Means Clustering, Probability Theory, Multivariate Gaussians, Performance
	101 (Colic)	Thursday	03-Feb	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	04-Feb	9:00-11:00	
	Both	*Project 1 Due*		*04-Feb*	*at 23:00*
Week 5	*Both*	*Reading assignment 4 Due*		*07-Feb*	*at 21:00*
	101 (Colic)	Tuesday	08-Feb	12:00-15:00	**Mathematical Foundation of Data Processing**
	102 (Aref)	Wednesday	09-Feb	9:00-12:00	Linear Algebra, Analytical Geometry and Transformations, Data Augmentation
	101 (Colic)	Thursday	10-Feb	13:00-15:00	Tutorial 2 - Anomaly Detection
	102 (Aref)	Friday	11-Feb	9:00-11:00	
Week 6	No lectures, no office hours				**Midterm Assessment: Feb 15 at 9:00 to Feb 16 at 15:00 (limited 2-hour window to start the exam and submit it)**
	101 (Colic)	Thursday	17-Feb	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	18-Feb	9:00-11:00	
Reading Week	*101 (Colic)*	Tuesday	22-Feb	12:00-15:00	No lectures and office hours during the reading week.
	102 (Aref)	Wednesday	23-Feb	9:00-12:00	
	101 (Colic)	Thursday	24-Feb	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	25-Feb	9:00-11:00	

Midterm

- Consists of multiple choice, short answer, math and programming questions as well as analytical and reasoning questions
- Cover all material before the midterm
- Late midterm submissions will receive a grade of 0
- Access to all course materials (except for Piazza)
- More details will be provided 1 – 2 weeks before the midterm

Tentative Schedule (Weeks 7 – 9)

Week 7	*Both*	*Project 2 Due*		*28-Feb*	*at 23:00*
	Both	*Reading assignment 5 Due*		*28-Feb*	*at 21:00*
	101 (Colic)	Tuesday	01-Mar	12:00-15:00	**Dimensionty Reduction Part 1**
	102 (Aref)	Wednesday	02-Mar	9:00-12:00	Projection, Matrix Decomposition, Eigenvectors, Principal Component Analysis
	101 (Colic)	Thursday	03-Mar	13:00-15:00	Tutorial 3 - PCA
	102 (Aref)	Friday	04-Mar	9:00-11:00	
Week 8	*Both*	*Reading assignment 6 Due*		*07-Mar*	*at 21:00*
	101 (Colic)	Tuesday	08-Mar	12:00-15:00	**Dimensionty Reduction Part 2**
	102 (Aref)	Wednesday	09-Mar	9:00-12:00	Singular Value Decomposition, Feature Interpretation, Vector Calculus
	101 (Colic)	Thursday	10-Mar	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	11-Mar	9:00-11:00	
	Both	*Project 3 Due*		*11-Mar*	*at 23:00*
Week 9	*Both*	*Reading assignment 7 Due*		*14-Mar*	*at 21:00*
	101 (Colic)	Tuesday	15-Mar	12:00-15:00	**Generalized Linear Model**
	102 (Aref)	Wednesday	16-Mar	9:00-12:00	Monte Carlo, Linear Regression, Gradient Descent, Polynomial Regression, Regularization
	101 (Colic)	Thursday	17-Mar	13:00-15:00	Tutorial 4 - Linear Regression
	102 (Aref)	Friday	18-Mar	9:00-11:00	

Tentative Schedule (Weeks 10 – 13)

Week 10	*Both*	*Reading assignment 8 Due*		*21-Mar*	*at 21:00*
	101 (Colic)	Tuesday	22-Mar	12:00-15:00	**Artificial Neural Networks**
	102 (Aref)	Wednesday	23-Mar	9:00-12:00	Continuous Optimization, Convexity, Classification, Perceptron, Neural Networks
	101 (Colic)	Thursday	24-Mar	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	25-Mar	9:00-11:00	
Week 11	*101 (Colic)*	Tuesday	29-Mar	12:00-15:00	**Deep Learning**
	102 (Aref)	Wednesday	30-Mar	9:00-12:00	Backward propagation, Deep Learning, Transfer Learning, Discrete Optimization
	101 (Colic)	Thursday	31-Mar	13:00-15:00	Q/A Support Session
	102 (Aref)	Friday	01-Apr	9:00-11:00	
	Both	*Project 4 Due*		*01-Apr*	*at 23:00*
Week 12	*101 (Colic)*	Tuesday	05-Apr	12:00-15:00	**Course Review**
	102 (Aref)	Wednesday	06-Apr	9:00-12:00	
	101 (Colic)	Thursday	07-Apr	13:00-15:00	No lab sessions on week 12.
	102 (Aref)	Friday	08-Apr	9:00-11:00	
Week 13	No lectures, no office hours, no lab sessions				**Final Assessment: Apr 12 at 9:00 to Apr 13 at 15:00 (limited 3-hour window to start the exam and submit it)**

Slide Attribution

The slides used for APS1070 contain materials from various sources. Special thanks to the following authors:

- Ali Hadi Zadeh
- Jason Riordon
- Sinisa Colic
- Mark C. Wilson (Lecture 2 in particular)

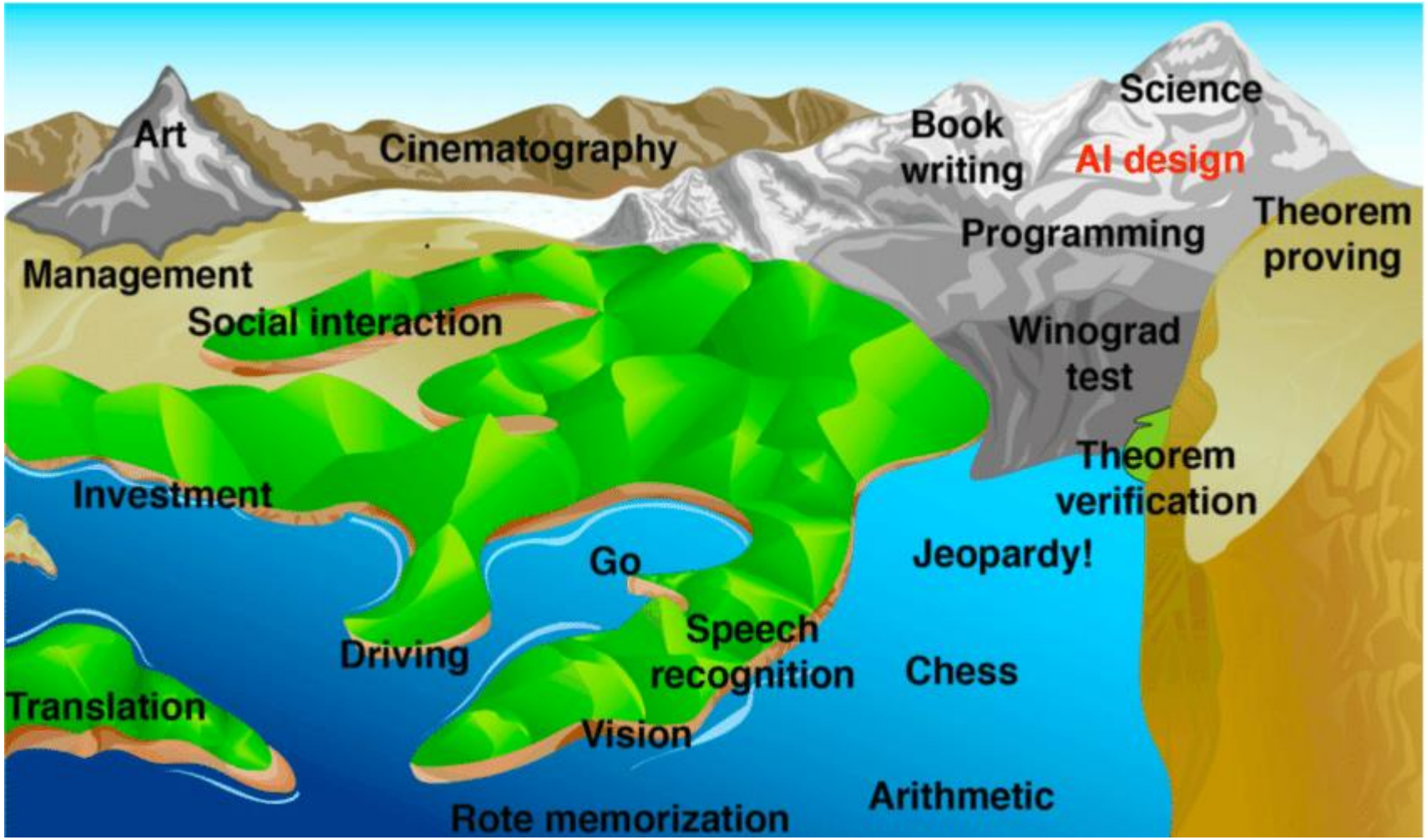
Part 2

Machine Learning Overview

Why are we here?

- Machine Learning skills is in demand
- Neural Networks are evolving and leading to new performance limits
- Deep Neural Networks are at the cutting-edge of applied computing

Rising Tide of AI Capacity



- Jobs requiring creativity seems to be a safe career choice…
- Programming and AI design seem safe

Source: Hans-Moravecs-rising-tide-of-AI-capacity

Why Machine Learning?

Q: How can we solve a programming problem?

- Evaluate all conditions and write a set of rules that efficiently address those conditions
  - ex. robot to navigate maze, turn towards opening

Q: How could we write a set of rules to determine if a goat is in the image?

Requires systems that can learn from examples…

Examples of Applications

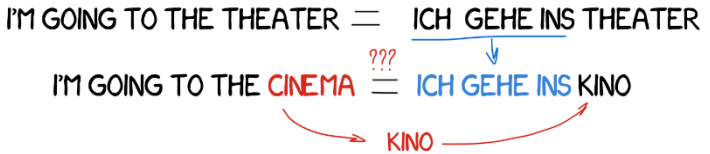
- Finance and banking
- Production
- Health
- Economics
- Data -> Knowledge -> Insight
- Marketing
- Intelligent assistants

Why Machine Learning?

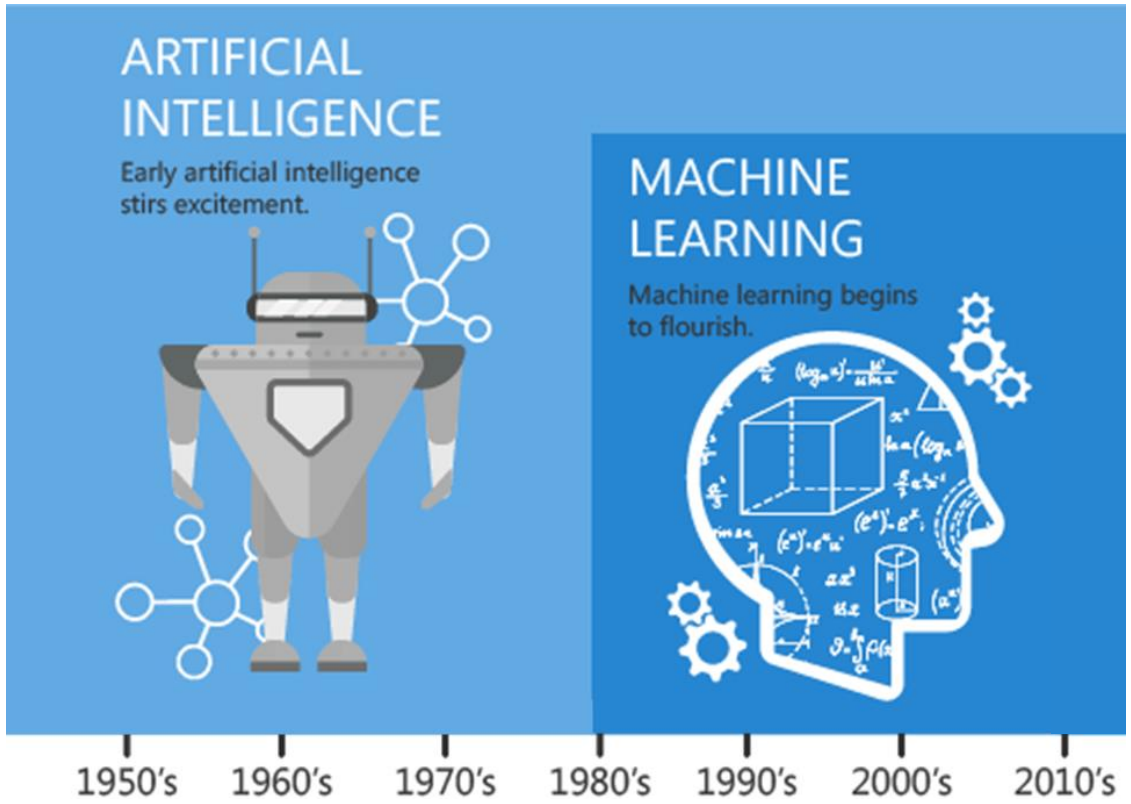
- Problems for which it is **difficult to formulate rules** that cover all the conditions that we are expected to see, or that require a lot of fine-turning.
- **Complex problems** where no good solutions exist, state-of-the-art Machine Learning techniques may be able to succeed.
- **Fluctuating environments**: a Machine Learning system can adapt to new data.
- **Obtaining insights** from large amount of data or complex problems.

Why Machine Learning?

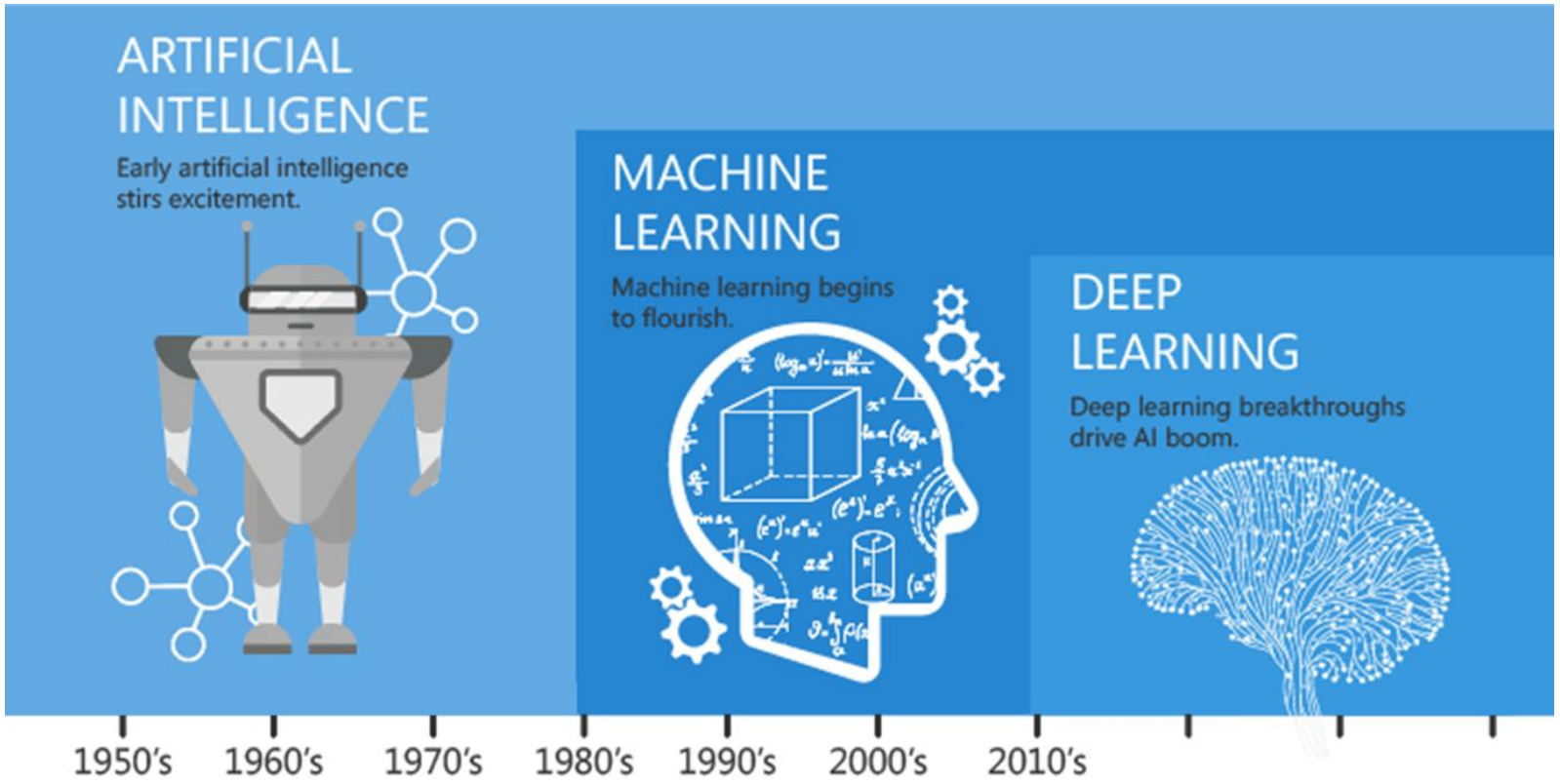
- A machine learning algorithm then takes “training data” and produces a model to generate the correct output
- If done correctly the program will generalize to cases not observed…more on this later
- **Instead of writing programs** by hand the **focus shifts to collecting quality examples** that highlight the correct output



What is Machine Learning?

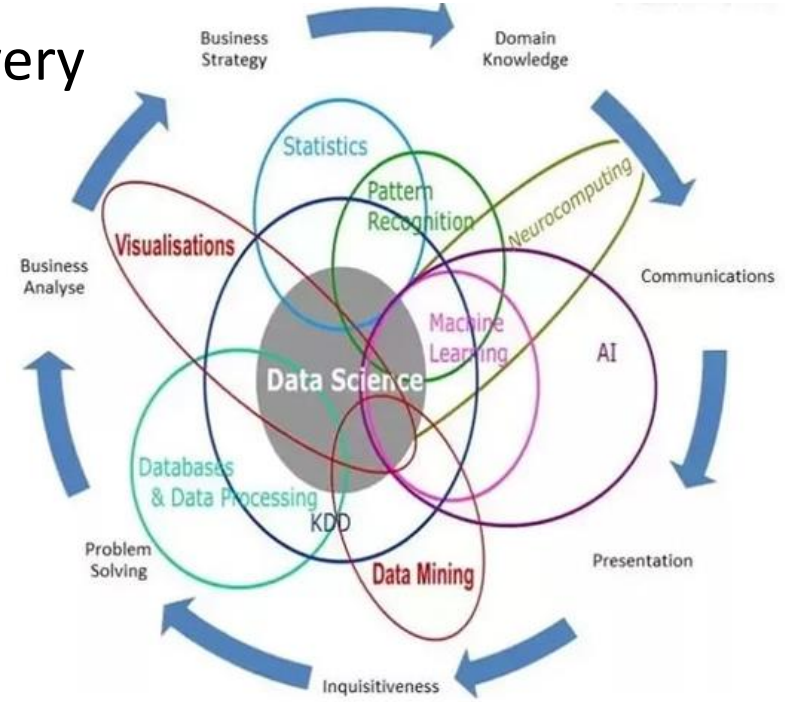


What is Machine Learning?

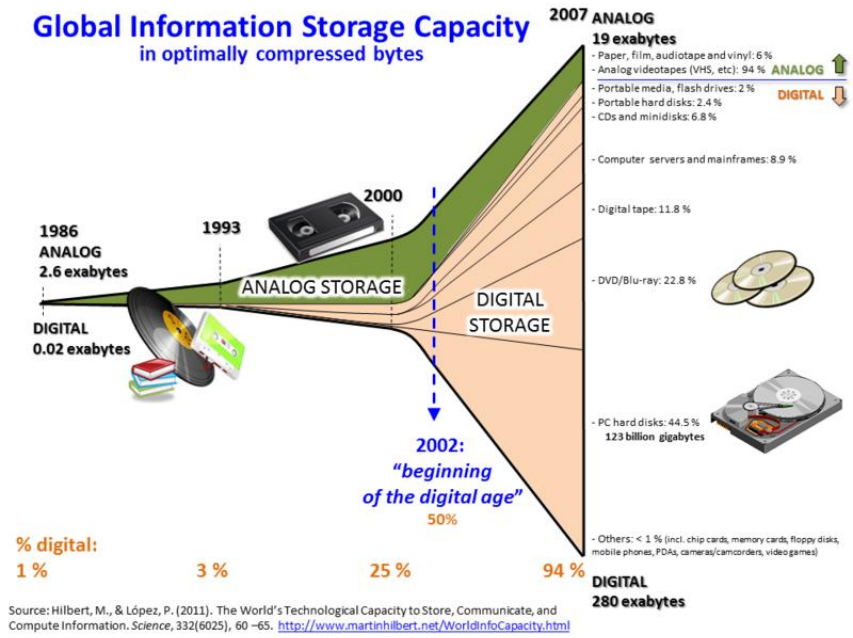


What is Data Science?

- Data Science
  - Multidisciplinary
  - Digital revolution
  - Data-driven discovery
- Includes:
  - Data Mining
  - Machine Learning
  - Big Data
  - Databases
  - …

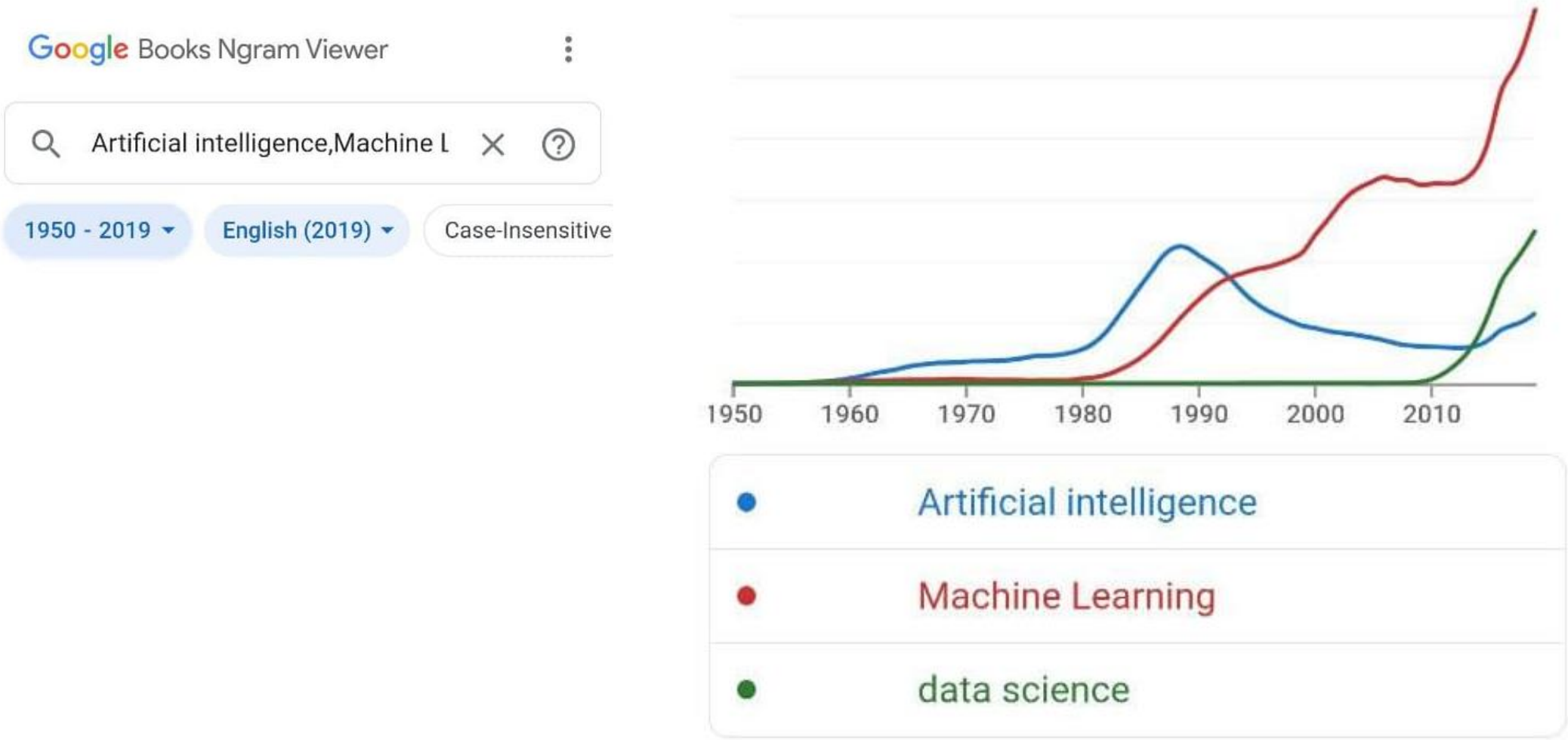


Source: Brendan Tierney, 2012



http://www.martinhilbert.net/

ML, AL, and Data Science over years



Types of Machine Learning Systems

It is useful to classify machine learning systems into broad categories based on the following criteria:

- supervised, unsupervised, semi-supervised, and reinforcement learning
- classification versus regression
- online versus batch learning
- instance-based versus model-based learning
- parametric or nonparametric

Supervised/Unsupervised Learning

- Machine Learning systems can be classified according to the amount and type of supervision they get during training.
  - Supervised
    - k-Nearest Neighbours, Linear Regression, Logistic Regression, Decision Trees, Neural Networks, and many more
  - Unsupervised
    - K-Means, Principal Component Analysis
  - Semi-supervised
  - Reinforcement Learning

Instance-Based/Model-Based Learning

- **Instance-Based:** system learns the examples by heart, then generalizes to new cases by using a similarity/distance measure to compare them to the learned examples.
  - more details in week 3, an example in today's lecture

- **Model-Based:** build a model of these examples and then use that model to make predictions.
  - more details in weeks 9 to 11

Challenges of Machine Learning

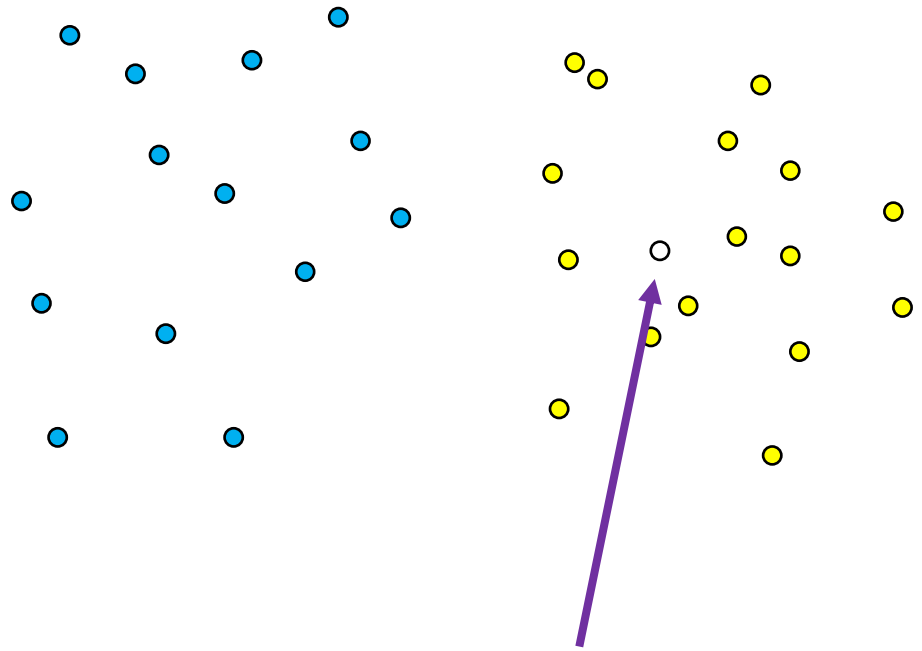
- Insufficient Data
- Quality Data
- Representative Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data
- Testing and Validation
- Hyperparameter Tuning and Model Selection
- Data Mismatch
- Fairness, Societal Concerns

Part 3

K-nearest neighbour classifier

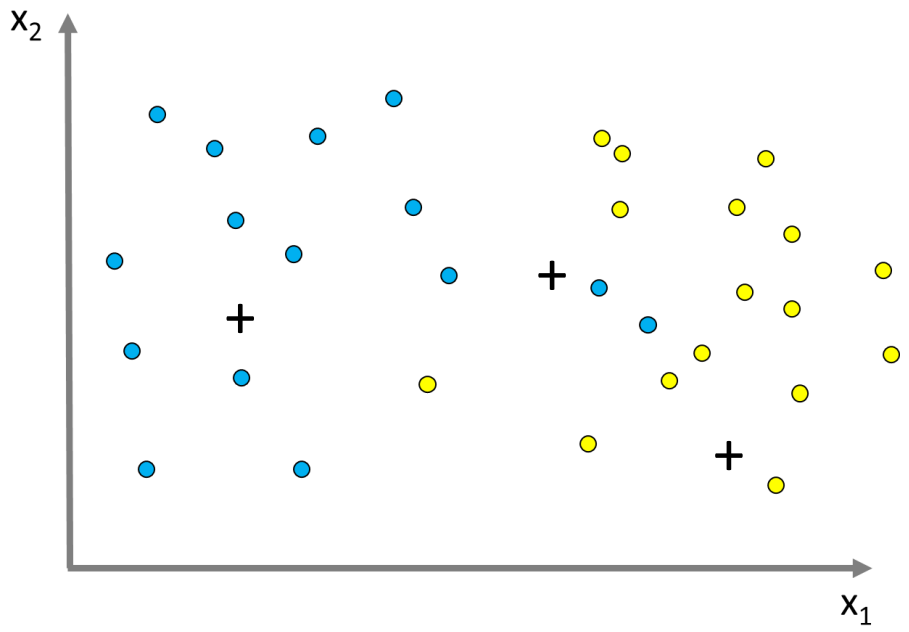
Nearest neighbour classifier

- Output is a class (here, blue or yellow)
- Instance-based learning, or lazy learning: computation only happens once called
- Flexible approach – no assumptions on data distribution

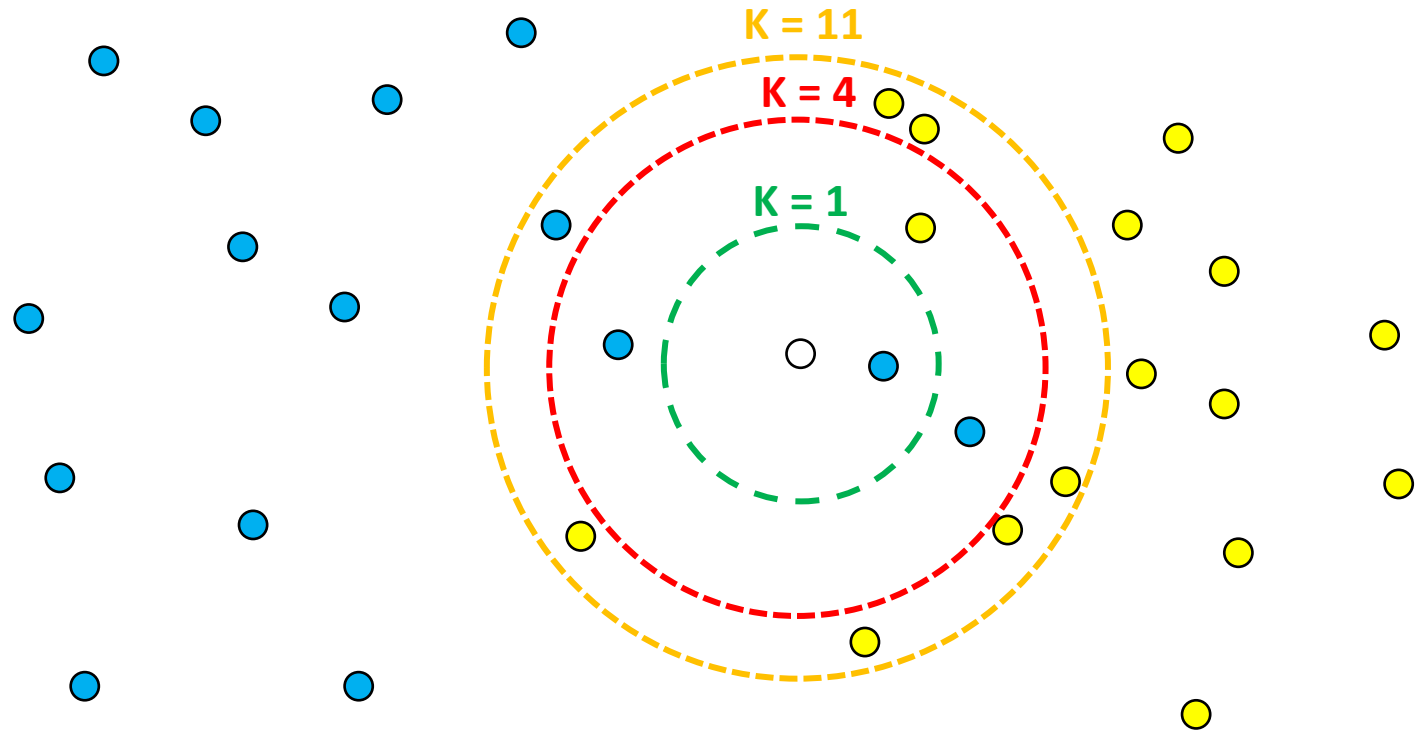


k-Nearest Neighbour Classification

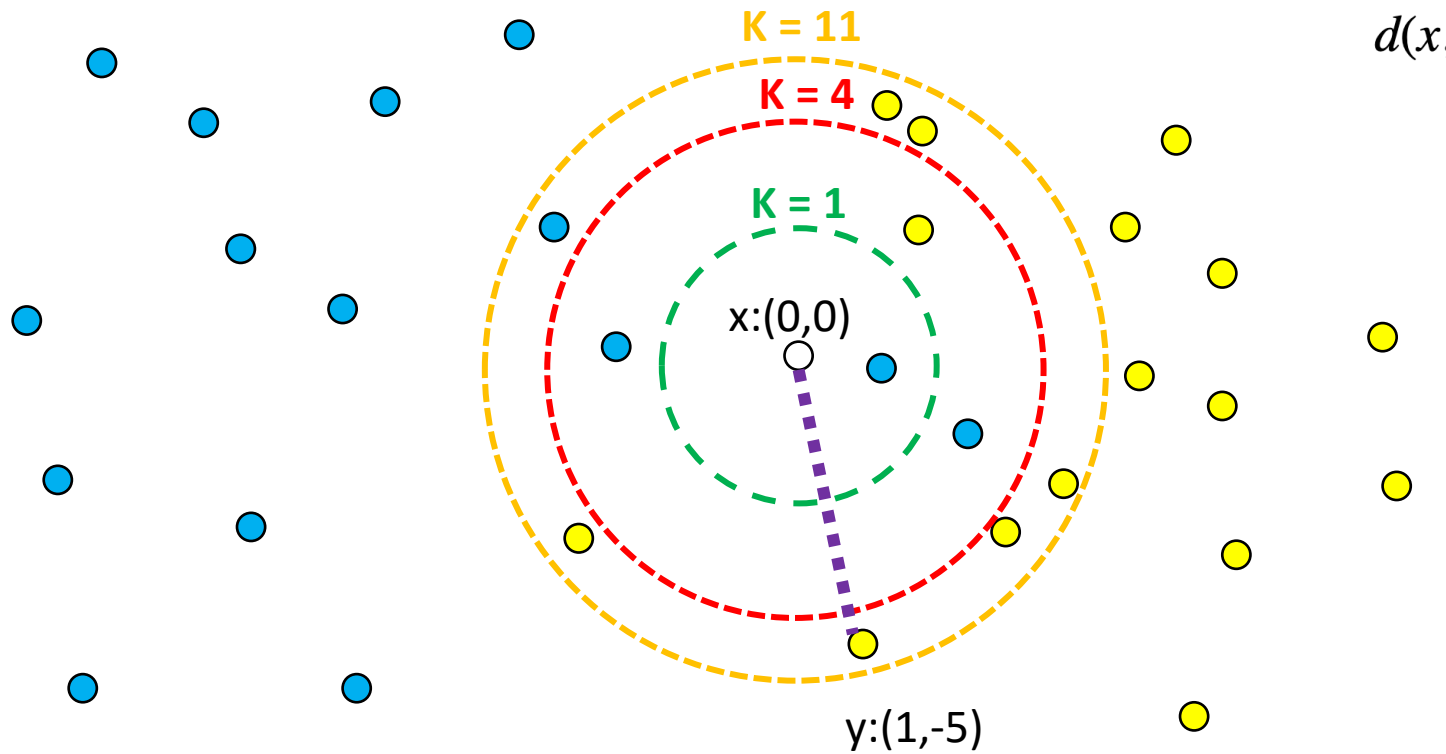
- Works on multidimensional data
- For visualization purposes we will use a 2-dimensional example
- + represents unknown test samples that we want to classify



K- nearest neighbour classifier



How to compute distance?



$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Euclidean distance

Simple Algorithm

- We can make the assumption that similar samples will be located close together
- Simple Algorithm

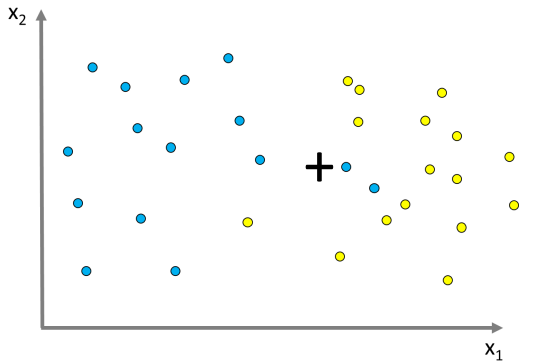
Calculate distance

Obtain the nearest neighbour

Determine labels

Algorithm

- For a single nearest neighbour



1. Find example $(\mathbf{x}^*, t^*)$ (from the stored training set) closest to $\mathbf{x}$. That is:

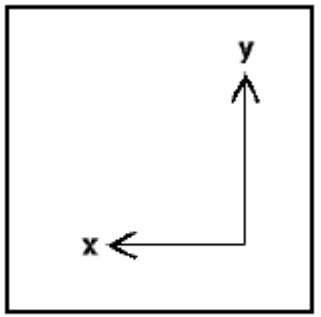
$$\mathbf{x}^* = \underset{\mathbf{x}^{(i)} \in \text{train. set}}{\text{argmin}} \ \text{distance}(\mathbf{x}^{(i)}, \mathbf{x})$$

2. Output $y = t^*$

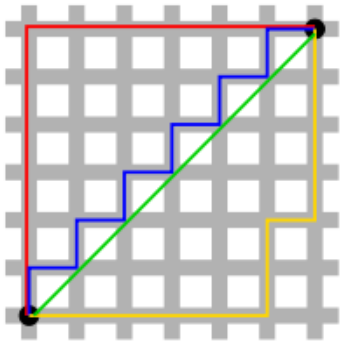
How do we measure distance?

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

p = 1, Manhattan Distance

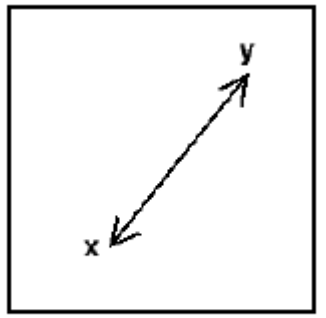


Manhattan



Source — Taxicab geometry Wikipedia

p = 2, Euclidean Distance

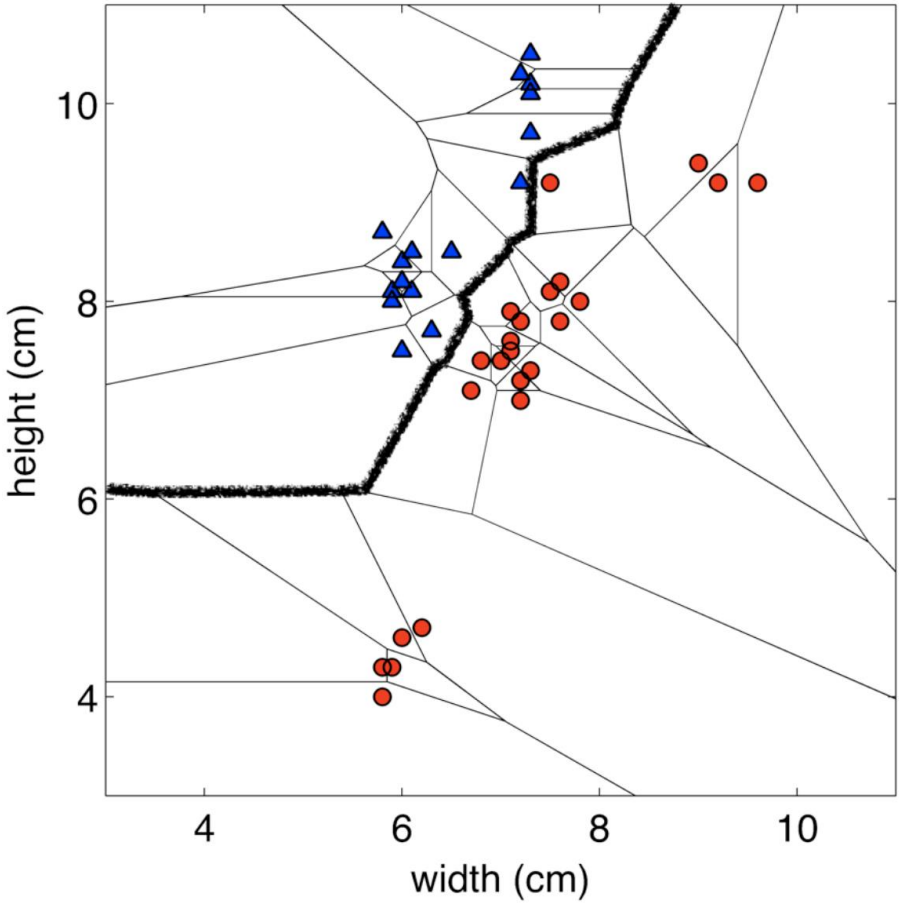


Euclidean

Dimension of x or y

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

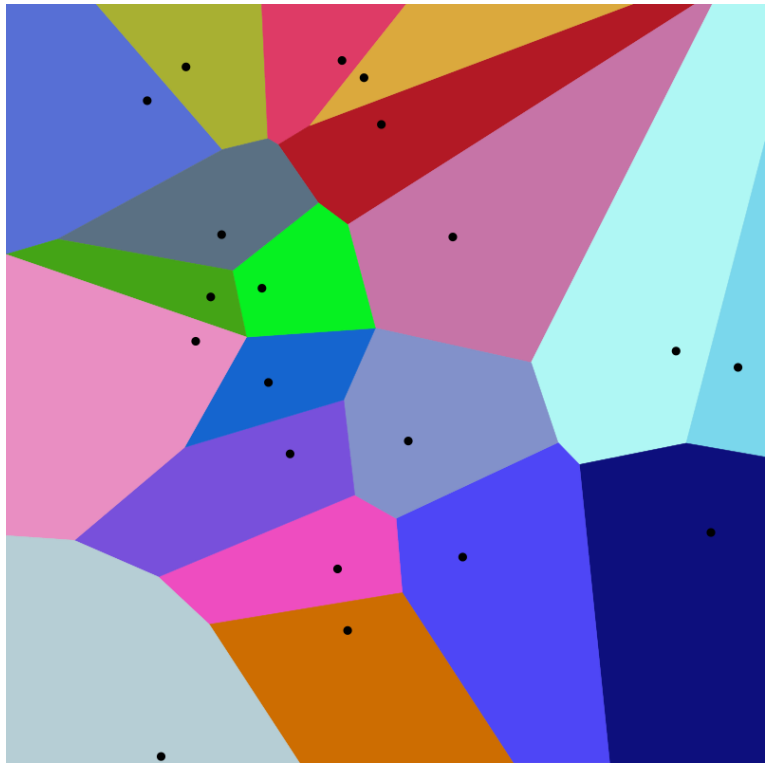
Decision Boundary



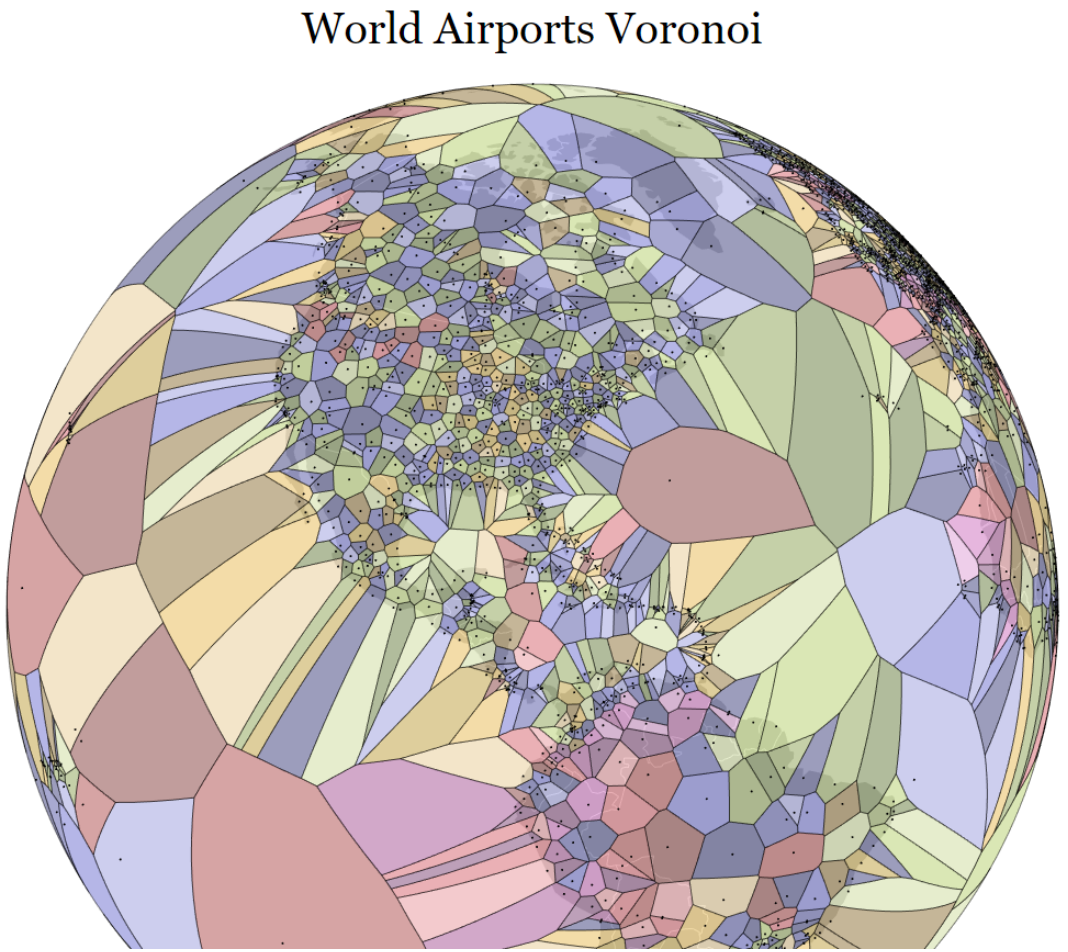
- Can generate arbitrary test points on the plane and apply kNN
- The boundary between regions of input space assigned to different categories.

Voronoi Diagrams (k=1)

Euclidian distance



Source: Wikipedia



Spherical Voronoi – Source: Jason Davies

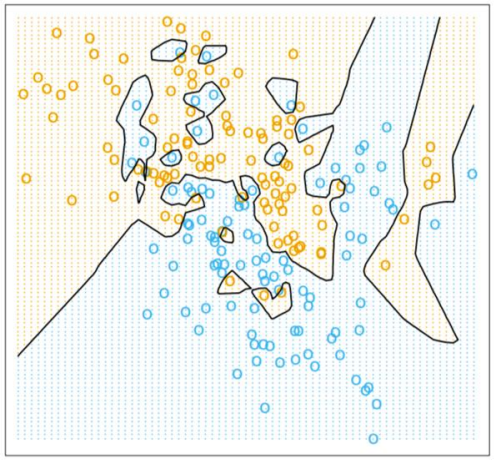
Selection of k

- Q: What happens if we let k be very small?
- Q: What happens if we let k be very large?

Tradeoffs in choosing k?

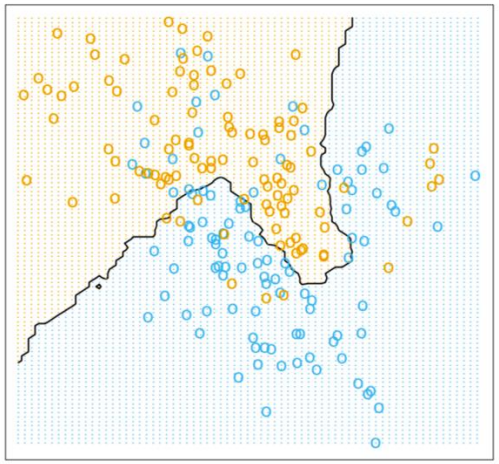
- **Small k**
  - Good at capturing fine-grained patterns
  - May overfit, i.e. be sensitive to random noise

Excellent for training data, not that good for new data, too complex



- **Large k**
  - Makes stable predictions by averaging over lots of examples
  - May underfit, i.e. fail to capture important regularities

Not that good for training data, not good for new data, too simple



What is the best k?

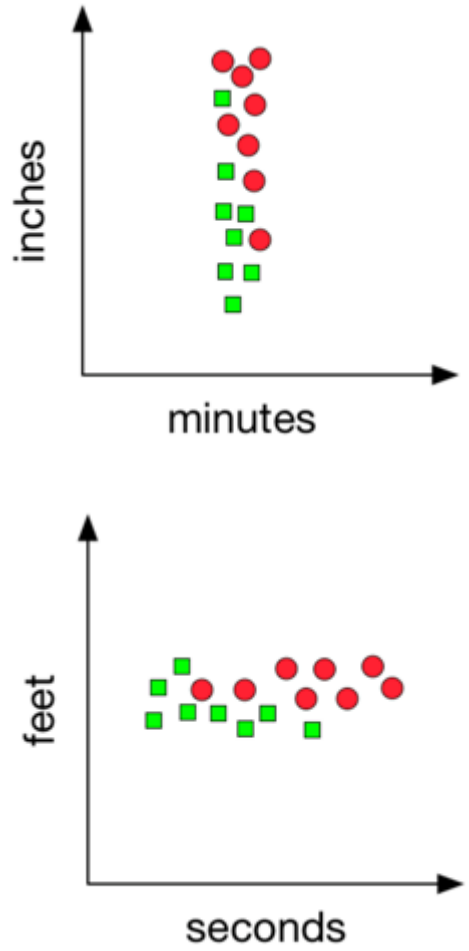
- Select the k based on the best performance on the validation set, report the results on the test set
- Generally, the performance on the validation set will be better than on the test set
- Q: How do you think it will perform on the training set?

Normalization

- Nearest Neighbours can be sensitive to the ranges of different features
- Often, the units are arbitrary
- **Simple fix:** normalize each dimension to be zero mean and unit variance (i.e., compute the mean μ_j and standard deviation σ_j, and take,

$$\tilde{x}_j = \frac{x_j - \mu_j}{\sigma_j}$$

- **Caution:** depending on the problem, the scale might be important!



K- nearest neighbour classifier

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

sklearn.neighbors.KNeighborsClassifier

class sklearn.neighbors.**KNeighborsClassifier**(*n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs*) [source]

Classifier implementing the k-nearest neighbors vote.

Read more in the User Guide.

Parameters:

n_neighbors : *int, default=5*
Number of neighbors to use by default for `kneighbors` queries.

weights : *{'uniform', 'distance'} or callable, default='uniform'*
weight function used in prediction. Possible values:

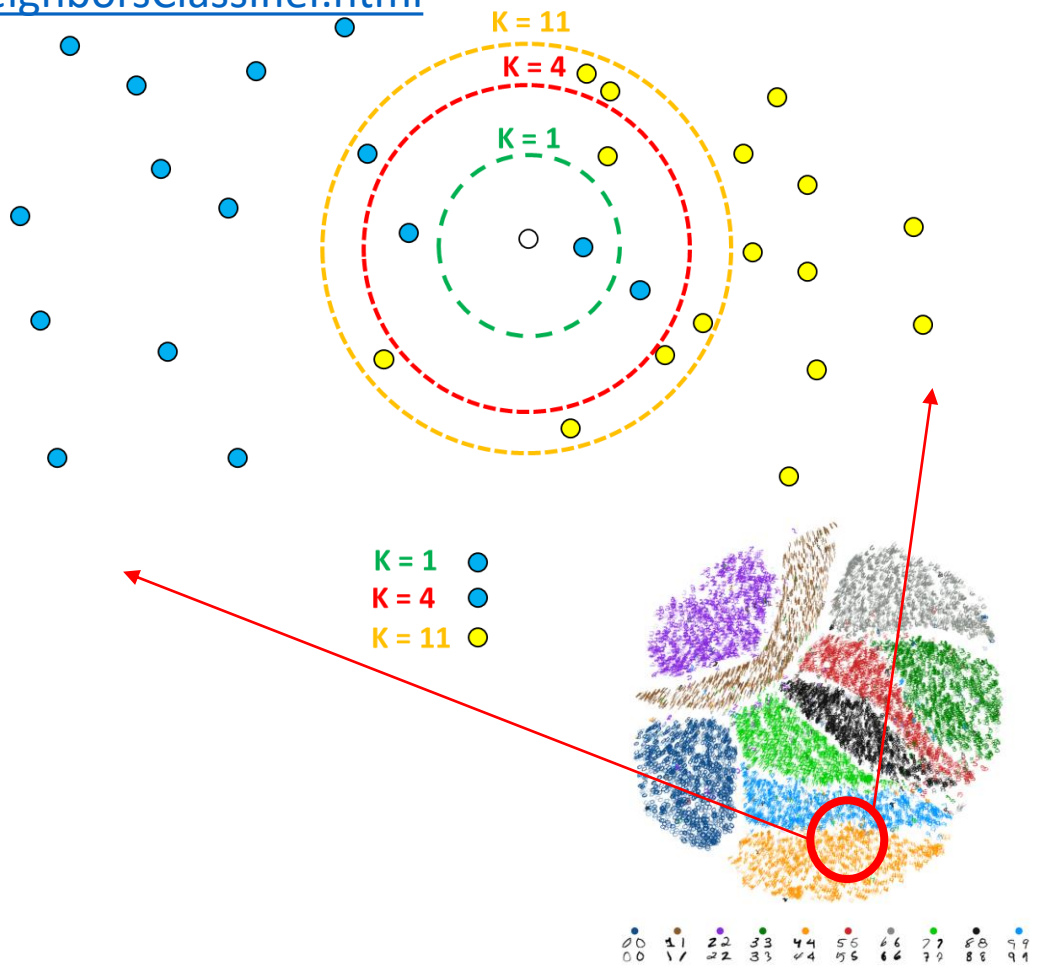
- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

algorithm : *{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'*
Algorithm used to compute the nearest neighbors:

- 'ball_tree' will use `BallTree`
- 'kd_tree' will use `KDTree`
- 'brute' will use a brute-force search.
- 'auto' will attempt to decide the most appropriate algorithm based on the values passed to `fit` method.

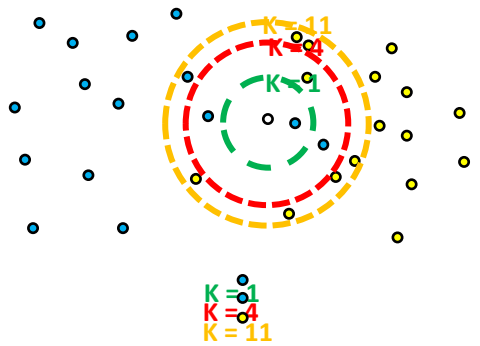
Note: fitting on sparse input will override the setting of this parameter, using brute force.

leaf_size : *int, default=30*



K- nearest neighbour classifier

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html



Examples

```
>>> X = [[0], [1], [2], [3]]
>>> y = [0, 0, 1, 1]
>>> from sklearn.neighbors import KNeighborsClassifier
>>> neigh = KNeighborsClassifier(n_neighbors=3)
>>> neigh.fit(X, y)
KNeighborsClassifier(...)
>>> print(neigh.predict([[1.1]]))
[0]
>>> print(neigh.predict_proba([[0.9]]))
[[0.66666667 0.33333333]]
```

Hide prompts and outputs

Methods

fit(X, y)	Fit the k-nearest neighbors classifier from the training dataset.
get_params([deep])	Get parameters for this estimator.
kneighbors([X, n_neighbors, return_distance])	Finds the K-neighbors of a point.
kneighbors_graph([X, n_neighbors, mode])	Computes the (weighted) graph of k-Neighbors for points in X
predict(X)	Predict the class labels for the provided data.
predict_proba(X)	Return probability estimates for the test data X.
score(X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.
set_params(**params)	Set the parameters of this estimator.

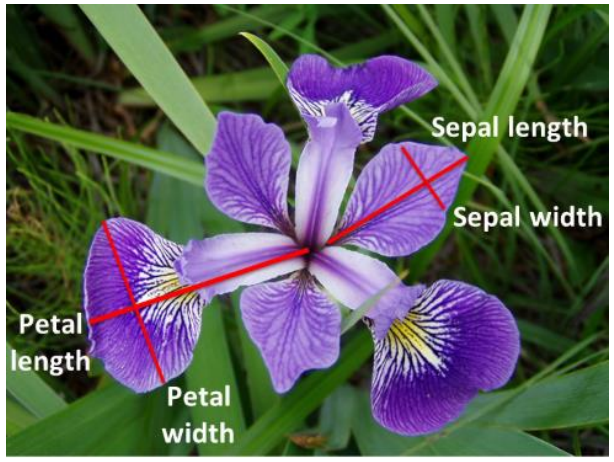
Important definitions

- Task : Flower classification
- Target (label, Class)
- Features

Iris setosa

Iris versicolor

Iris virginica



Important definitions

- Task : Flower classification
- Target (label, Class) : Setosa, Versicolor, Virginica.
- Features: Petal len, Petal wid, Sepal len, Sepal wic
- Model
- Prediction
- Data point (sample)
- Dataset

Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2

Features

Fisher's *Iris* data [hide]

Dataset order	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	*I. setosa*
2	4.9	3.0	1.4	0.2	*I. setosa*
3	4.7	3.2	1.3	0.2	*I. setosa*
4	4.6	3.1	1.5	0.2	*I. setosa*
5	5.0	3.6	1.4	0.3	*I. setosa*
6	5.4	3.9	1.7	0.4	*I. setosa*
7	4.6	3.4	1.4	0.3	*I. setosa*
8	5.0	3.4	1.5	0.2	*I. setosa*
9	4.4	2.9	1.4	0.2	*I. setosa*
10	4.9	3.1	1.5	0.1	*I. setosa*
11	5.4	3.7	1.5	0.2	*I. setosa*
12	4.8	3.4	1.6	0.2	*I. setosa*
13	4.8	3.0	1.4	0.1	*I. setosa*
14	4.3	3.0	1.1	0.1	*I. setosa*
15	5.8	4.0	1.2	0.2	*I. setosa*
16	5.7	4.4	1.5	0.4	*I. setosa*
17	5.4	3.9	1.3	0.4	*I. setosa*
18	5.1	3.5	1.4	0.3	*I. setosa*
19	5.7	3.8	1.7	0.3	*I. setosa*
20	5.1	3.8	1.5	0.3	*I. setosa*
21	5.4	3.4	1.7	0.2	*I. setosa*
22	5.1	3.7	1.5	0.4	*I. setosa*
23	4.6	3.6	1.0	0.2	*I. setosa*
24	5.1	3.3	1.7	0.5	*I. setosa*
25	4.8	3.4	1.9	0.2	*I. setosa*
26	5.0	3.0	1.6	0.2	*I. setosa*
27	5.0	3.4	1.6	0.4	*I. setosa*
28	5.2	3.5	1.5	0.2	*I. setosa*
29	5.2	3.4	1.4	0.2	*I. setosa*
30	4.7	3.2	1.6	0.2	*I. setosa*
31	4.8	3.1	1.6	0.2	*I. setosa*
32	5.4	3.4	1.5	0.4	*I. setosa*
33	5.2	4.1	1.5	0.1	*I. setosa*
34	5.5	4.2	1.4	0.2	*I. setosa*
35	4.9	3.1	1.5	0.2	*I. setosa*
36	5.0	3.2	1.2	0.2	*I. setosa*
37	5.5	3.5	1.3	0.2	*I. setosa*
38	4.9	3.6	1.4	0.1	*I. setosa*
39	4.4	3.0	1.3	0.2	*I. setosa*
40	5.1	3.4	1.5	0.2	*I. setosa*
41	5.0	3.5	1.3	0.3	*I. setosa*
42	4.5	2.3	1.3	0.3	*I. setosa*
43	4.4	3.2	1.3	0.2	*I. setosa*

Prediction:

Versicolor

th	Petal width	Species
	0.2	*I. setosa*

target

KNN Code Example (Google Colab)

Next Time

- Reading assignment 1 Due - Jan. 17 at 21:00
  - Read Pages 15-20 of Chapter 1 from “Introduction to Algorithms and Data Structures”, 4th Ed, by Michael J. Dinneen, Georgy Gimel’farb, and Mark C. Wilson, 2016 link
  - Complete a quiz on Quercus and submit it by the deadline
- Week 1 Lab on Thursday or Friday: Tutorial 0
  - Python Basics
- Week 2 Lecture – Analysis of Algorithms
  - Algorithms and Big O Notation
  - Sorting
  - Hashing
- Project 1 Due - Feb. 4 at 23:00