Some images from this lecture are taken from Google Image Search, contact me if you want the reference
Big Data Phenomenon

• We are **collecting and storing data** at an unprecedented rate.

• Examples:
  – YouTube, Facebook, MOOCs, news sites.
  – Credit cards transactions and Amazon purchases.
  – Transportation data (Google Maps, Waze, Uber)
  – Gene expression data and protein interaction assays.
  – Maps and satellite data.
  – Large hadron collider and surveying the sky.
  – Phone call records and speech recognition results.
  – Video game worlds and user actions.
Big Data Phenomenon

• What do you do with all this data?
  – Too much data to search through it manually.

• But there is valuable information in the data.
  – How can we use it for fun, profit, and/or the greater good?

• Data mining and machine learning are key tools we use to make sense of large datasets.
Data Mining

• Automatically extract useful knowledge from large datasets.

• Usually, to help with human decision making.
Machine Learning

• Using computer to automatically detect patterns in data and use these to make predictions or decisions.

• Most useful when:
  – We want to automate something a human can do.
  – We want to do things a human can’t do (look at 1 TB of data).
Data Mining vs. Machine Learning

• Data mining and machine learning are very similar:
  – Data mining often viewed as closer to databases.
  – Machine learning often viewed as closer AI.

• Both are similar to statistics, but more emphasis on:
  – Large datasets and computation.
  – Predictions (instead of descriptions).
  – Flexible models (that work on many problems).
Deep Learning vs. Machine Learning vs. AI

• Traditional we’ve viewed ML as a subset of AI.
  – And “deep learning” as a subset of ML.
Applications

• Spam filtering:

• Credit card fraud detection:

• Product recommendation:
Applications

• Motion capture:

• Optical character recognition and machine translation:

• Speech recognition:
Applications

• Face detection:

• Object detection:

• Sports analytics:
Applications

• Personal Assistants:

• Medical imaging:

• Self-driving cars:
Applications

• Scene completion:

• Image annotation:
Applications

• Discovering new cancer subtypes:

• Automated Statistician:

2.4 Component 4: An approximately periodic function with a period of 10.8 years. This function applies until 1643 and from 1716 onwards.

This component is approximately periodic with a period of 10.8 years. Across periods, the shape of this function varies smoothly with a typical lengthscale of 36.9 years. The shape of this function within each period is very smooth and resembles a sinusoid. This component applies until 1643 and from 1716 onwards.
Applications

• Mimicking artistic styles and inceptionism:
Applications

• “Deep dream”:
Applications

• Fast physics-based animation:

• Mimicking art style in video.
• Recent work on generating text/music/voice/poetry/dance.
Applications

• Beating human Go masters:
• **Summary:**
  – There is a lot you can do with a bit of statistics and a lot data/computation.

• **But it is important to know the limitations of what you are doing.**
  – “The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.” – John Tukey
  – A huge number of people applying ML are just “overfitting”.

• **We are in exciting times.**
  – Major recent progress in fields like speech recognition and computer vision.
  – Things are changing a lot on the timescale of 3-5 years.
  – A bubble in ML investments.
(pause)
Reasons NOT to take this class

• Compared to typical CS classes, there is a lot more math:
  – Requires linear algebra, probability, and multivariate calculus (at once).
  – “I think the prerequisites for this course should require that students have obtained at least 75% (or around there) in the required math courses. As someone who did not excel at math, I felt severely under prepared and struggled immensely in this course, especially seeing that I have taken CPSC courses in the past with similar math requirements, but were not nearly as math heavy as CPSC340.”

• If you’ve only taken a few math courses (or have low math grades), this course will ruin your life for the next 4 months.

• It’s better to improve your math, then take this course later.
  – Take MATH 302 or 307 instead.
Reasons NOT to take this class

• This is not a class on “how to use scikit-learn or TensorFlow”.
  – You will need to implement things from scratch, and modify existing code.

• Instead, this is a 300-level computer science course:
  – You are expected to be able to quickly understand and write code.
  – You are expected to be able to analyze algorithms in big-O notation.

• We’re going to use the Julia programming language.
  – You are expected to be able to learn a programming language on your own.
  – Mike Gelbart teaches it in Python.

• If you only have limited programming experience, this course will ruin your life for the next 4 months.

• It’s better to get programming experience, then take this course later.
  – Take CPSC 310 or 320 instead.
Reasons NOT to take this class

• Do NOT take this grade expecting a high grade with low effort.

• Many people find the assignments very long and very difficult.
  – You will need to put time and effort into learning new/difficult skills.
  – If you aren’t strong at math and CS, they may take all of your time.

• Class averages have only been high because of graduate students.
  – NOT because this is an “easy” course, it’s not.
CPSC 340 vs. CPSC 540

• There is also a graduate ML course, CPSC 540:
  – More advanced material.
  – More focus on theory/implementation, less focus on applications.
  – More prerequisites and higher workload.

• For almost all students, CPSC 340 is the right class to take:
  – CPSC 340 focuses on the most widely-used methods in practice.
    • It covers much more material than standard ML classes like Coursera.
  – CPSC 540 focuses on less widely-used methods and research topics.
    • It is intended as a continuation of CPSC 340.
    • You’ll miss important topics if you skip CPSC 340.
Essential Links

• Please bookmark the course homepage:
  – Contains lecture slides, assignments, optional readings, additional notes.

• You should sign up for Piazza:
  – Can be used to ask questions about lectures/assignments/exams.
  – May occasionally be used for course announcements.

• Use Piazza instead of e-mail for questions:
  – I can take a long time to respond e-mails.
Textbooks

• No required textbook.

• I’ll post relevant sections out of these books as optional readings:
  – Introduction to Data Mining (Tan et al.).
  – The Elements of Statistical Learning (Hastie et al.).
  – Mining Massive Datasets (Leskovec et al.)
  – Machine Learning: A Probabilistic Perspective (Murphy).

• Most of these are on reserve in the ICICS reading room.
• List of related courses on the webpage, or you can use Google.
Assignments and Working in Teams

• There will be 6 Assignments worth 30% of final grade:
  – Usually a combination of math and programming.
  – Submitted as a zip file using the Handin program.
    • You will need to setup a CS account to use this.
  – Make sure to follow the formatting instructions (hand in early and often).

• Assignment 0 is on the webpage, and is due next Friday.

• Assignment 0 must be done individually.
• Assignments 1-5 can be done in pairs.
  – There is no commitment to keep the same pairs between assignments.
Late “Class” Policy for Assignments

• Assignments will be due at midnight “anytime on Earth” (ATE).

• If you can’t make it, you can use “late classes”:
  – For example, if assignment is due on a Friday:
    • Handing it in Friday is 0 late classes.
    • Handing it in Monday is 1 late class.
    • Handing it in Wednesday is 2 late classes.

  – You will get a mark of 0 on an assignment if you:
    • Use more than 2 late classes on the assignment.
    • Exceed 4 late classes across all assignments.
    • Submit the solutions to an assignment from a previous term.

• We’ll try to put grades on Connect within 10 days of due date.
Programming Language: Julia

• 3 most-used languages in these areas: Python, Matlab, and R.

• We will be using Julia which is similar to Matlab.
  – Except it’s free and is way faster than Python/Matlab/R.

• No, you cannot use Python/Matlab/R/etc.
  – Assignments have prepared code that we won’t translate to 3 languages.
  – TAs shouldn’t have to know 3 languages to grade.
Waiting List and Auditing

- Right now only CS students register directly.
- 181/195 seats are filled, but the room supports 250 students.

- We’re going to start registering people from the waiting list.
  - Being on the waiting list is the only way to get registered:
    - https://www.cs.ubc.ca/students/undergrad/courses/waitlists
  - You might be registered without being notified, be sure to check!
    - They might also ask to submit a prereq form, let me know if you have issues.

- Because the room is full, we may not have seats for auditors.
  - If there is space, I’ll describe (light) auditing requirements then.
Getting Help

• Many students find the assignments long and difficult.
• But there are many sources of help:
  – TA office hours and instructor office hours (see webpage for times).
    • Starting in the second week of class.
  – Piazza.
  – Weekly tutorials.
    • Starting in second week of class.
    • Will go through provided code, review background material, review big concepts, and/or do exercises.
    • Tutorials are optional be you must be registered in a tutorial section to stay enrolled.
  – Other students (ask your neighbor for their e-mail).
  – The web (almost all topics are covered in many places).
Midterm and Final

• In-class midterm worth 20% and a (cumulative) final worth 50%
  – Closed-book.
  – One doubled-sided ‘cheat sheet’ for midterm.
  – Two doubled-sided pages for final.
  – No need to pass the final to pass the course (but recommended).

• Midterm is tentatively schedule for October 20\textsuperscript{th}.
• I don’t control when the final is, don’t make travel plans before December 22\textsuperscript{nd}.

• There will be two types of questions:
  – ‘Technical’ questions requiring things like pseudo-code or derivations.
    • Similar to assignment questions, only be related topics covered in assignments.
  – ‘Conceptual’ questions testing understanding of key concepts.
    • All lecture slide material except “bonus slides” is fair game here.
Lectures

• All slides will be posted online (before lecture, and final version after).

• Please ask questions: you probably have similar questions to others.
  – I may deflect to the next lecture or Piazza for certain questions.

• Be warned that the **course we will move fast and cover a lot of topics:**
  – Big ideas will be covered slowly and carefully.
  – But a bunch of other topics won’t be covered in a lot of detail.

• Isn’t it wrong to have only have shallow knowledge?
  – In this field, it’s **better to know many methods** than to know 5 in detail.
    • This is called the “no free lunch” theorem: different problems need different solutions.
• I will include a lot of “bonus slides”.
  – May mention advanced variations of methods from lecture.
  – May overview big topics that we don’t have time for.
  – May go over technical details that would derail class.

• You are **not expected to learn** the material on these slides.
  – But they’re useful if you want to take 540 or work in this area.

• I’ll use this colour of background on bonus slides.
Code of Conduct

• Do not post offensive or disrespectful content on Piazza.
• If you have a problem or complaint, let me know (maybe we can fix it).
• Do not distribute any course materials without permission.
• Do not record lectures without permission.

• Think about how/when to ask for help:
  – Don’t ask for help after being stuck for 10 seconds. Make a reasonable effort to solve your problem (check instructions, Piazza, and Google).
  – But don’t wait until the 10th hour of debugging before asking for help.
    • If you do, the assignments will take all of your time.

• There will be no post-course grade changes based on grade thresholds:
  – 49% will not be rounded to 50%, and 71% will not be rounded to 72%.
Cheating and Plagiarism

• Read about UBC’s policy on “academic misconduct” (cheating):

• When submitting assignments, **acknowledge all sources**:
  – Put “I had help from Sally on this question” on your submission.
  – Put “I found this from another course’s material” on your submission.
  – Put “I copied this section from this website” on your submission.
  – Otherwise, this is **plagiarism** (course material/textbooks are ok with me).

• At Canadian schools, this is taken very seriously.
  – Could receive 0 in course, be expelled from UBC, or have degree revoked.
Course Outline

• Next class discusses data “exploratory data analysis”.

• After that, the remaining lectures focus on five topics:
  1) Supervised Learning.
  2) Unsupervised learning.
  3) Linear prediction.
  4) Latent-factor models.
  5) Deep learning.
(pause)
Supervised Learning

• **Classification:**
  – Given an object, assign it to predefined ‘classes’.

• **Examples:**
  – Spam filtering.
  – Body part recognition.
Unsupervised Learning

- **Clustering:**
  - Find groups of ‘similar’ items in data.

- **Examples:**
  - Are there subtypes of tumors?
  - Are there high-crime hotspots?

- **Outlier detection:**
  - Finding data that doesn’t belong.

- **Association rules:**
  - Finding items frequently ‘bought together’.
Linear Prediction

- **Regression:** Predicting continuous-valued outputs.
- **Working with very high-dimensional data.**
Latent-Factor Models

• Principal component analysis and friends:
  – Low-dimensional representations.
  – Decomposing objects into “parts”.
  – Visualizing high-dimensional data.

• Collaborative filtering:
  – Predicting user ratings of items.
Deep Learning

- **Neural networks**: Brain-inspired ML when you have a lot of data/computation but don’t know what is relevant.
Photo I took in the UK on the way home from the “Optimization and Big Data” workshop: