bioinformatics.ca

Canadian Bioinformatics Workshops

www.bioinformatics.ca

bioinformaticsdotca.github.io


Creative Commons
This page is available in the following languages:
Afrikaans Български Català Dansk Deutsch Ελληνικά English English (CA) English (GB) English (US) Esperanto
Castellano Castellano (AR) Español (CL) Castellano (CO) Español (Ecuador) Castellano (MX) Castellano (PE)
Euskara Suomeksi français français (CA) Galego עברית hrvatski Magyar Italiano 日本語 한국어 Macedonian Melayu
Nederlands Norsk Sesotho sa Leboa polski Português română slovenski jezik српски srpski (latinica) Sotho svenska
中文 華語 (台灣) isiZulu
creative commons
Attribution-Share Alike 2.5 Canada
You are free:
to Share — to copy, distribute and transmit the work
to Remix — to adapt the work
Free Cultural
APPROVED FOR
Works
Under the following conditions:
Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar licence to this one.
For any reuse or distribution, you must make clear to others the licence terms of this work.
Any of the above conditions can be waived if you get permission from the copyright holder.
The author's moral rights are retained in this licence.
Disclaimer
Your fair dealing and other rights are in no way affected by the above.
This is a human-readable summary of the Legal Code (the full licence) available in the following languages:
English French
Learn how to distribute your work using this licence

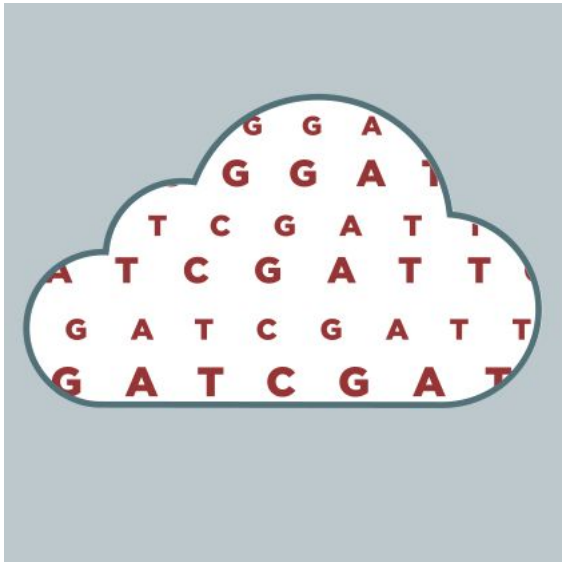
Module 1: Introduction to Machine Learning

David Wishart

Machine Learning for Bioinformatics

Aug. 16-17, 2023

bioinformatics.ca



Schedule

Day 1		Day 2	
Time (EDT)	Wednesday Aug. 16	Time (EDT)	Thursday Aug. 17
10:00	Introductions and Technology Check	10:00	Module 5: Gene Finding with NNs (Lecture and Lab)
10:45	Module 1: Introduction to Machine Learning (Lecture)	11:30	Break (30 min)
12:15	Break (30 min)	12:00	Module 6: Machine Learning with Keras and Scikit-Learn (Lecture/Lab)
12:45	Module 2: Decision Trees (Lecture and Lab)	14:00	Break (1 hour)
14:15	Break (45 min)	15:00	Module 7: Machine Learning with Keras and Scikit-Learn (Cont'd)
15:00	Module 3: Neural Networks (Lecture and Lab)	16:00	Break (30 min)
16:30	Break (30 min)	16:30	Module 8: Information Extraction with ChatGPT (Lecture and Lab)
17:00	Module 4: Neural Networks for 2^o Structure (Lecture and Homework)	17:45	Survey and Closing Remarks
18:00	End of Day 1	18:00	End of Day 2

How to Ask Questions

- The software Slack will be used for answering your questions throughout the course
- When you have a question or require clarification, please post to the slack group “mle”
- Your TA’s will respond to your thread directly
- Should you require more one-on-one attention, or if you have a question that a number of others also have, one of your TA’s will welcome you to a Zoom breakout room to discuss either one-on-one or in small groups

Learning Objectives

- To temper expectations
- To introduce, define and differentiate the concept of Machine Learning (ML)
- To show everyday examples of ML
- To show examples of ML in bioinformatics and/or genomics
- To explain the standard ML workflow
- To introduce the class to Google Colab, the class website & code repository

Tempering Expectations

- Machine learning is a powerful technique and is widely used in many fields
- Machine learning is a difficult subject with many sub-disciplines that take years to master and it usually requires advanced training in math, CS and statistics
- A 2-day course will not make you an expert in machine learning
- This is an introductory course to give you a "taste" of ML and hopefully to inspire you to learn more on your own

Programming Languages and Environment

- We will mainly be using Python on Google Collaboratory for this course
- Supplementary R code for each module will be available
- All code for the course will be provided
- No need to install python or a python environment

Learning vs. Machine Learning

- **Learning**
  - *any process by which an organism or system improves performance from experience*
- **Machine learning**
  - *a branch of artificial intelligence (AI) in which a computer automatically improves performance from experience*
  - *technique to develop programs that can make predictions or decisions without being explicitly programmed to do so*

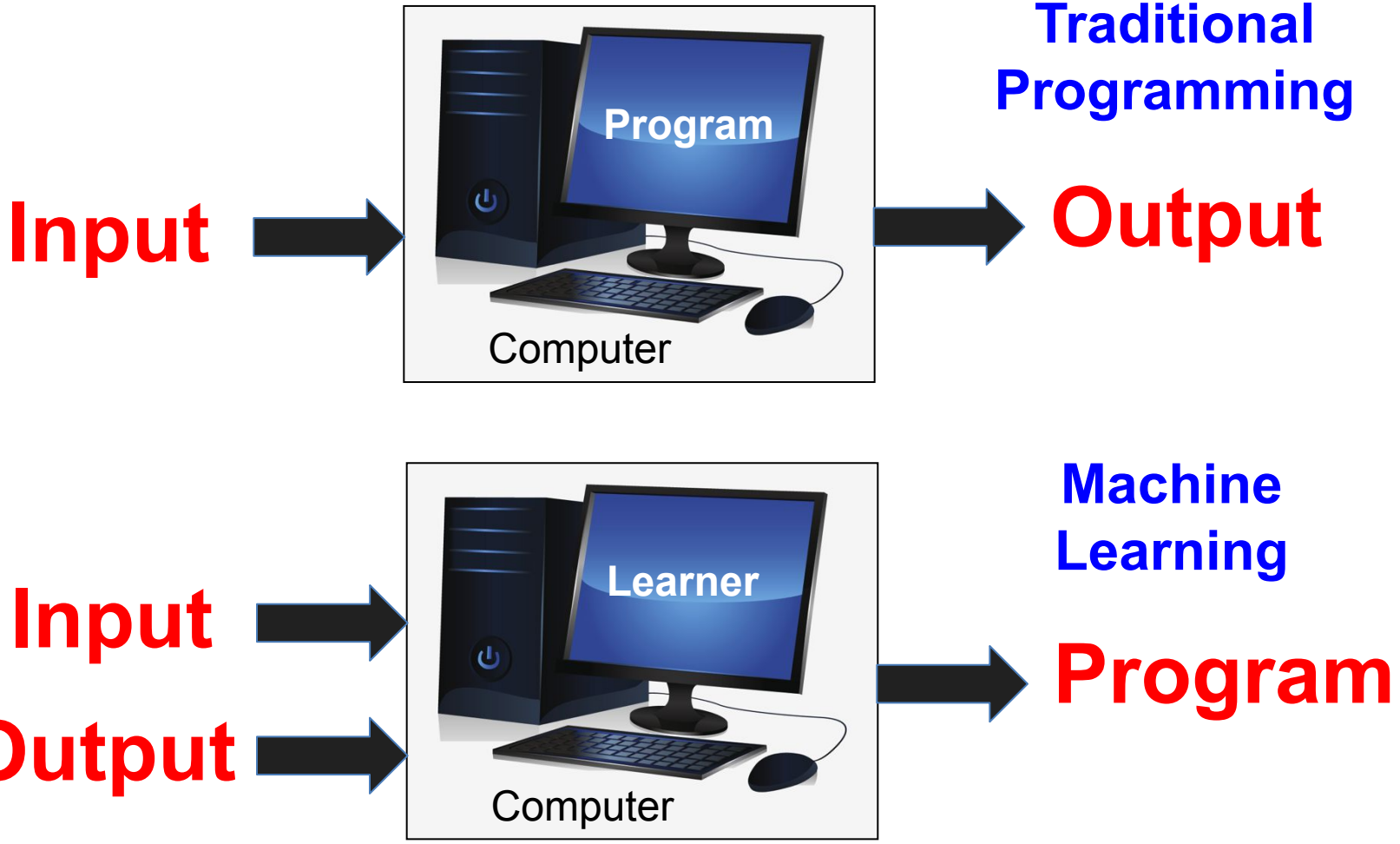
Machine Learning Is Old

"Machine Learning: The field of study that gives computers the ability to learn without being explicitly programmed."

- Arthur Samuel (1956)

Arthur Samuel playing checkers with the IBM 701

Machine Learning vs. Traditional Programming



Machine Learning vs. Traditional Programming *cont...*

Traditional Programming

1 + 1 → **= 2**

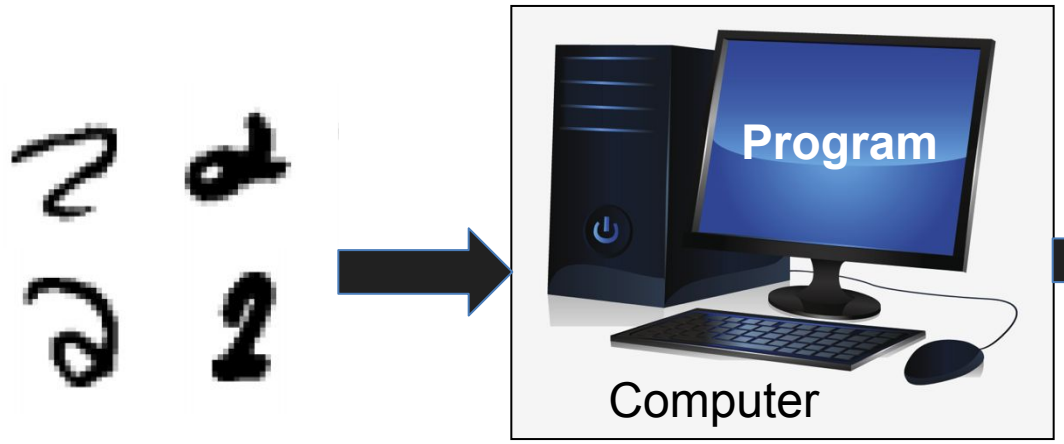
Lots of Examples

1 + 0 = 1	2 + 0 = 2
1 + 1 = 2	2 + 1 = 3
1 + 2 = 3	2 + 2 = 4
1 + 3 = 4	2 + 3 = 5
1 + 4 = 5	2 + 4 = 6
1 + 5 = 6	2 + 5 = 7
1 + 6 = 7	2 + 6 = 8

Machine Learning

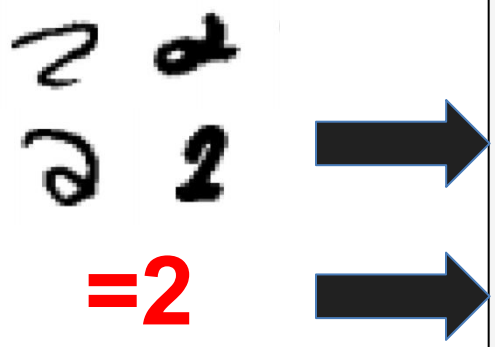
Model for Addition

Machine Learning vs. Traditional Programming *cont...*



Traditional Programming

= ?



=2

Machine Learning

Model for "2" recognition

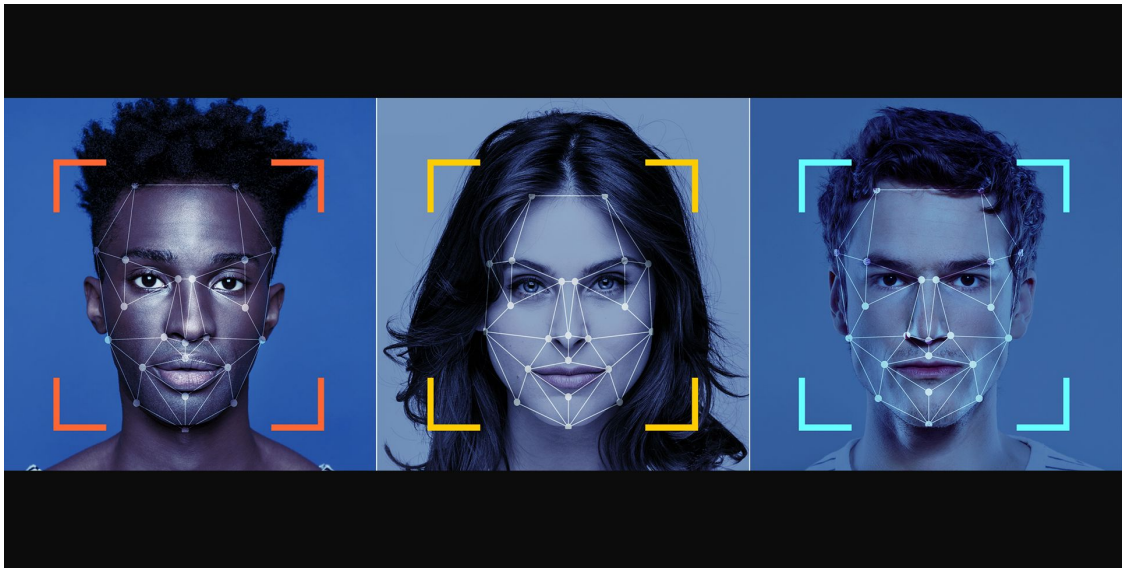
Machine Learning (ML) vs. Conventional Computing

- Conventional computer programs or algorithms perform tedious tasks faster and more accurately than humans (addition, subtraction, spell-checking)
- Machine learning algorithms perform tasks that are difficult or infeasible to do via conventional computer algorithms (grammar checking, interpreting speech, image recognition)

Machine Learning (ML) vs. Artificial Intelligence (AI)

- AI is the older field and many view ML as a subfield of AI – both require lots of data to really work
- Some view AI as different than ML
- AI has a focus on expert systems using defined rules, look-up tables and vast quantities of data to solve problems (chess)
- ML doesn't use expert systems, it uses probabilistic computing, computational statistics and optimization to try to make predictions (face recognition)
- Many people working in the field of ML started in AI, many now do both

ML vs. AI *cont...*



ML – facial recognition on cell phone cameras - 2020

AI – Deep Blue (IBM) vs. Gary Kasparov World Championship Chess - 1997

AI + ML & Jeopardy

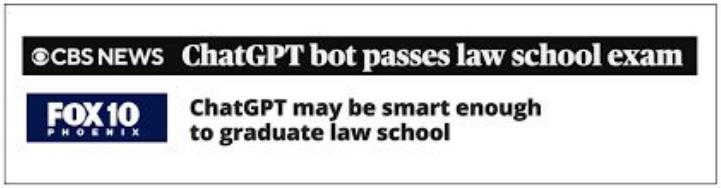
IBM – Watson vs Ken Jennings vs. Brad Rutter, 2011

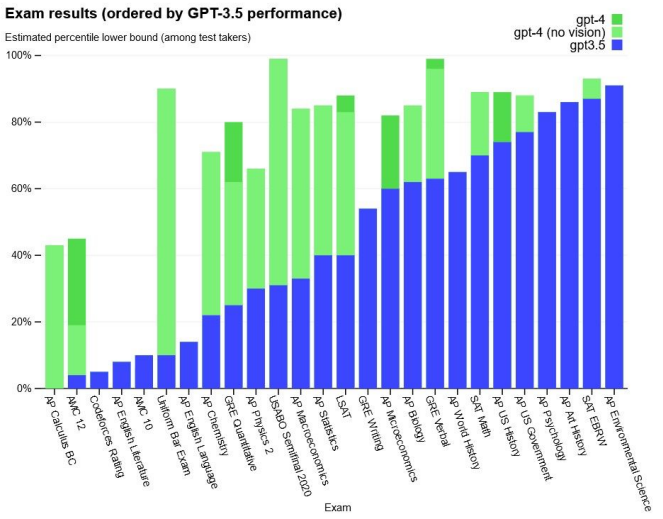
Watson (IBM)

- First developed in 2010 as a question answering (QA) system (took 4 years, 1000+ people, $900 million)
- Won Jeopardy ($1 million) in 2011
- Uses advanced natural language processing, information retrieval, knowledge representation, automated reasoning (all AI) and machine learning
- Sources of information for Watson include encyclopedias, dictionaries, thesauri, newswire articles, Dbpedia, WordNet and Yago
- Watson (on the Cloud) can now 'see', 'hear', 'read', 'talk', 'taste', 'interpret', 'learn' and 'recommend'

ChatGPT (OpenAI/Microsoft)

- Chat bot released by OpenAI in Nov. 30, 2022 (v. 3.5)
- Cost >$700 million to develop, 350 people working for 4 years
- GPT = Generative Pre-trained Transformer model (similar to a GNN)
- Large Language Model (LLM) based on 45 Tbytes of text or about 300 million pages or 500 billion words
- Essentially an extended version of auto-fill for texting
- Has passed numerous difficult exams (SAT, GRE, LSAT)





5 Cool Things To Do with ChatGPT

Create Original and Hilarious Jokes

Find Mistakes in a Code

Write codes for any problem

Write a Personalized Cover Letter or Resume

Write an Original Song

Machine Learning vs. Data Mining

- Both fields require lots of data, both can be used to predict
- ML focuses on reproducing or predicting from known knowledge
- Data mining focuses on discovery of previously unknown knowledge

Machine Learning vs. Deep Learning *cont...*

- Deep learning is a subdiscipline of ML
- Deep learning uses artificial neural networks (ANNs), specifically multi-layered "deep" neural networks (DNNs) as the main learning model – this allows DNNs to learn more complex patterns, handle tougher problems and make smarter predictions
- Other common ANN architectures include recurrent neural networks (RNNs), convolutional neural networks (CNNs) and deep belief networks (DPNs)

Key Players in Deep Learning

Geoffery Hinton – University of Toronto
Born in 1947, Educated at Edinburgh
Considered one of the fathers of ANNs
Won the 2018 Turing award
FRS, FRSC, CC
Now works with Google and UofT

Yoshua Bengio – University of Montreal
Born in 1964, Educated at McGill
Considered one of the godfathers of DL
Won the 2018 Turing award
FRS, FRSC, OC
Co-founder of Element AI

Three Approaches to ML

- **Supervised Learning (most common)**
  - Given example inputs and desired or labeled outputs with the goal being to learn rules that map inputs to outputs
- **Unsupervised Learning**
  - Given unlabeled data try to learn rules to find structure in the input data
- **Reinforcement Learning**
  - Learning to solve a problem by being given continuous feedback to maximize rewards

Three Approaches to ML

- **Supervised Learning (most common)**
  - Useful for classification and regression
  - Used in email filters, ranking/recommending systems, face verification, voice recognition
- **Unsupervised Learning**
  - Useful for pattern finding and clustering
  - Used in target recognition, seismic data analysis
- **Reinforcement Learning**
  - Useful in man-machine-environment applications
  - Used in game playing, autonomous car driving

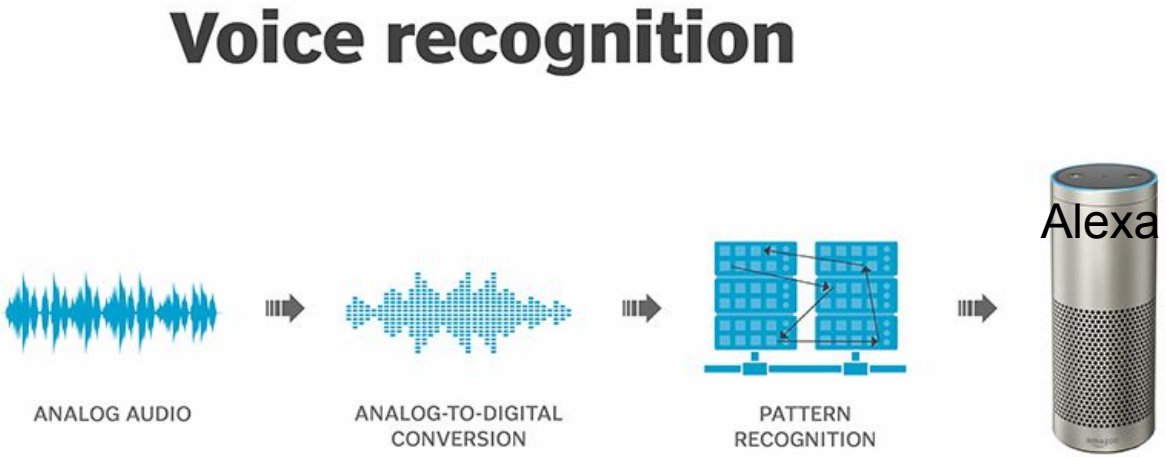
ML Uses Different "Models"

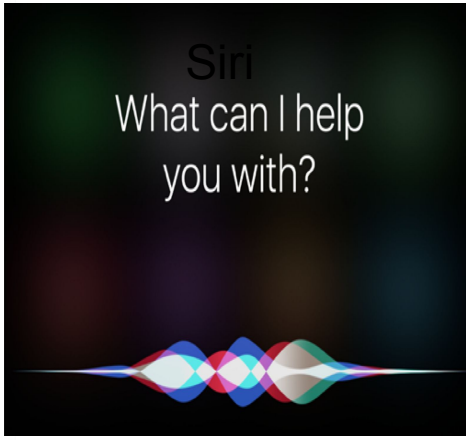
- All Machine Learning approaches require the use or creation of certain kinds of models
- Common models include:
  - Decision Trees/Random Forest
  - Artificial Neural Networks (ANNs)
  - Support Vector Machines (SVMs)
  - Genetic Algorithms (GAs)
  - Bayesian Networks (BNs)
  - Convolutional Neural Networks (CNNs)

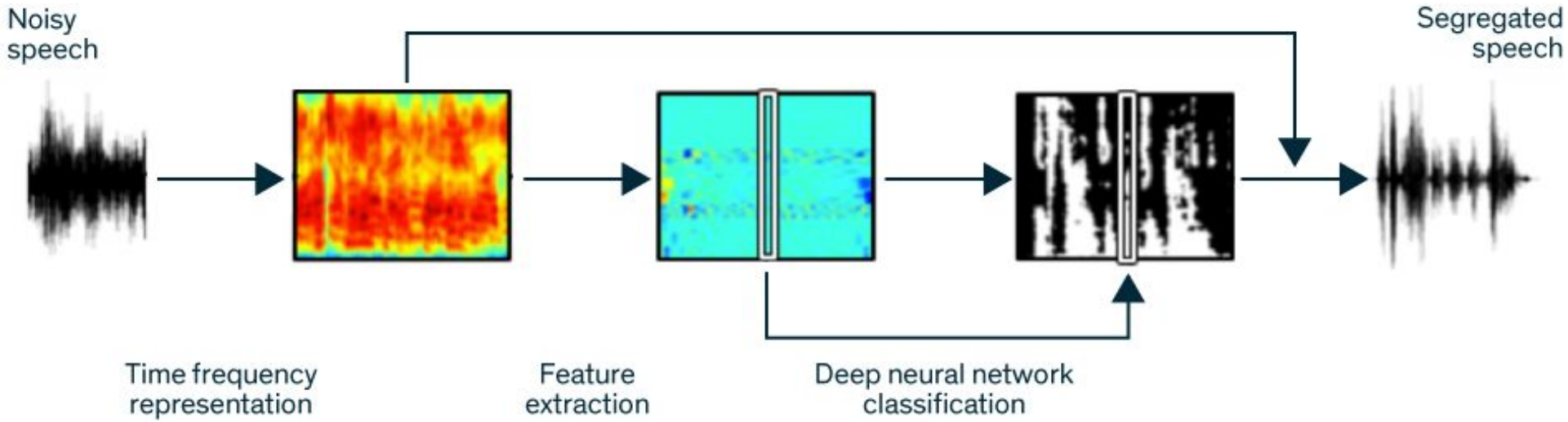
Machine Learning Applications

- Self-driving cars
- Game playing
- Face recognition
- Fingerprint recognition
- Traffic sign recognition
- Automated trading
- E-mail spam filtering
- Gesture recognition
- Speech recognition
- Handwriting recogn
- Radar analysis
- Resource management
- Brain simulation
- Medical diagnosis
- Bioinformatics

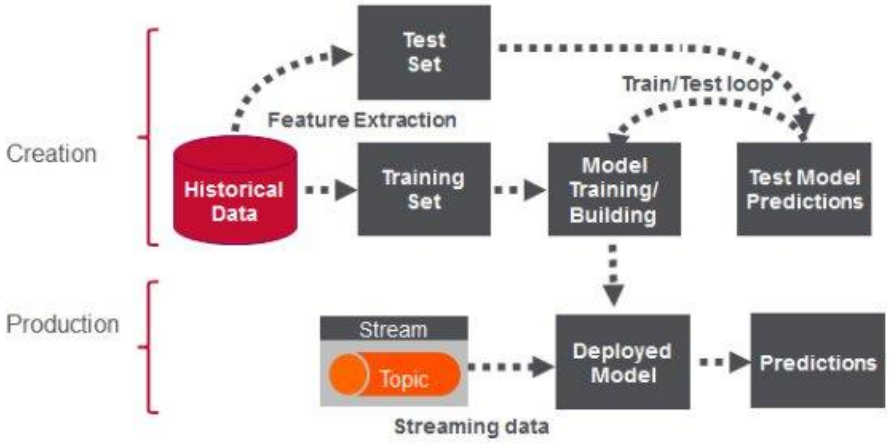
ML for Speech Recognition

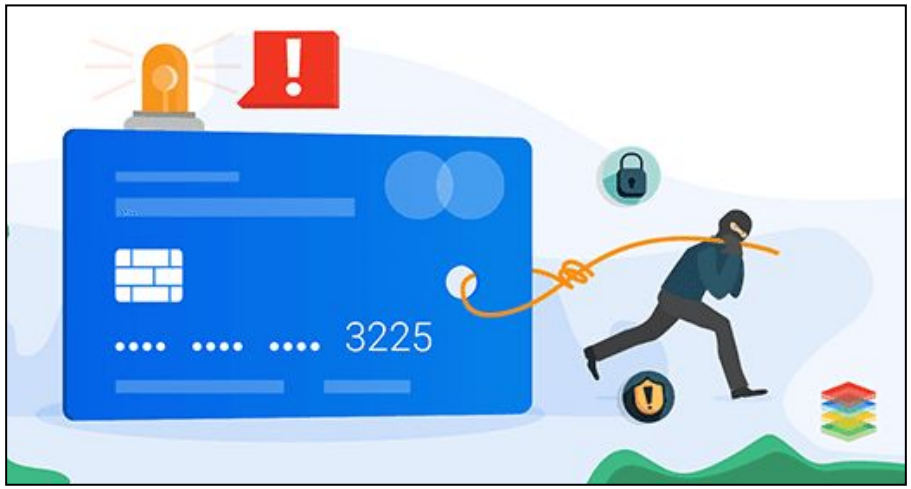






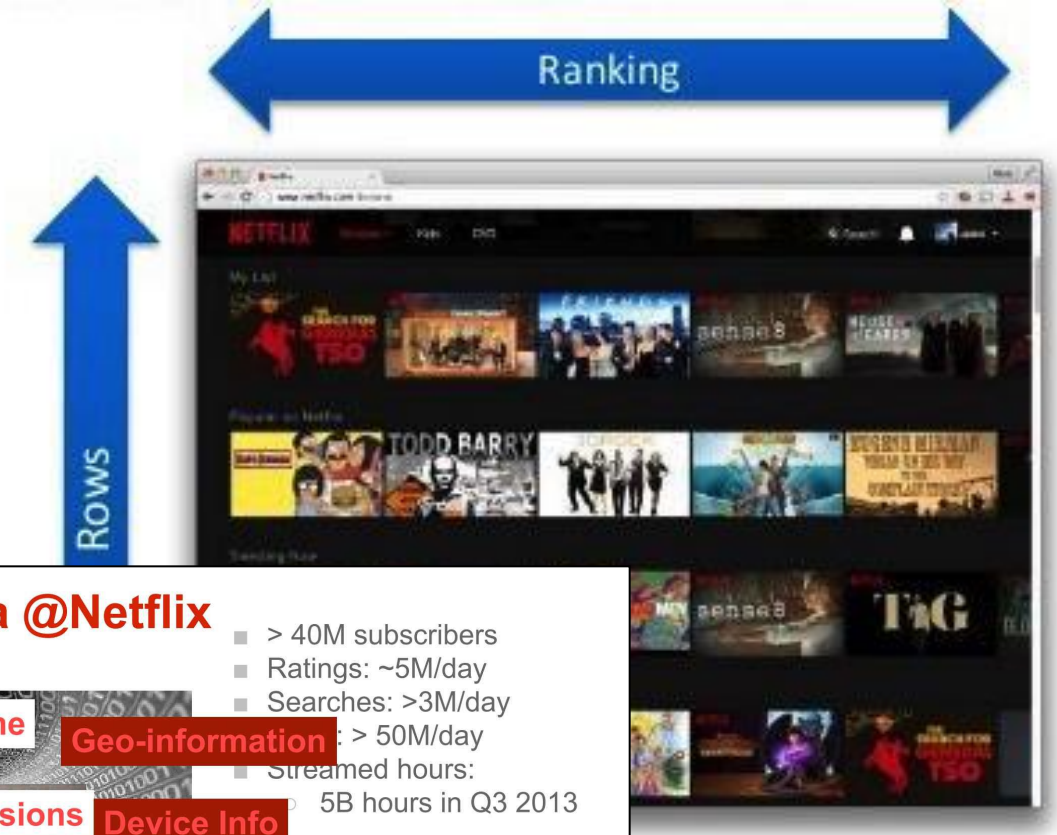
ML For Fraud Prevention





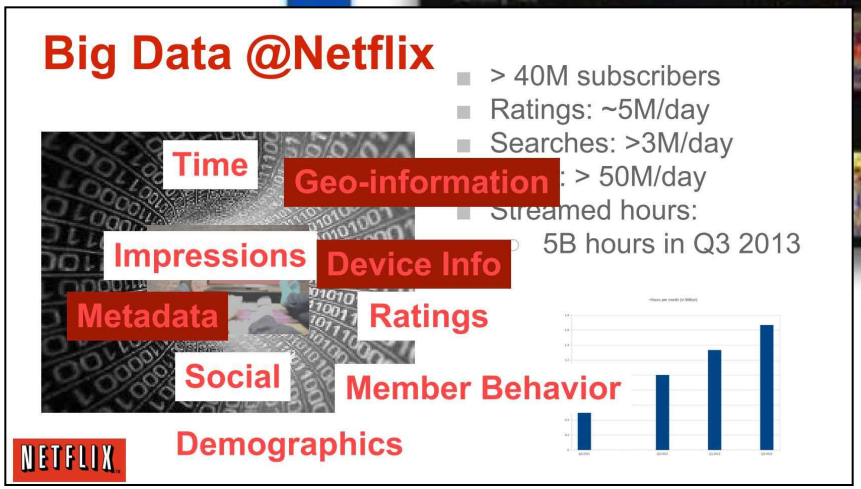
- Anomaly detection (unusual purchases, unusual locations)
- Historical purchase data (data mining)
- Pattern extraction from past purchase data sets
- Integrates user profile data (age, sex, job)
- Highly personalized

ML & Netflix Recommendations



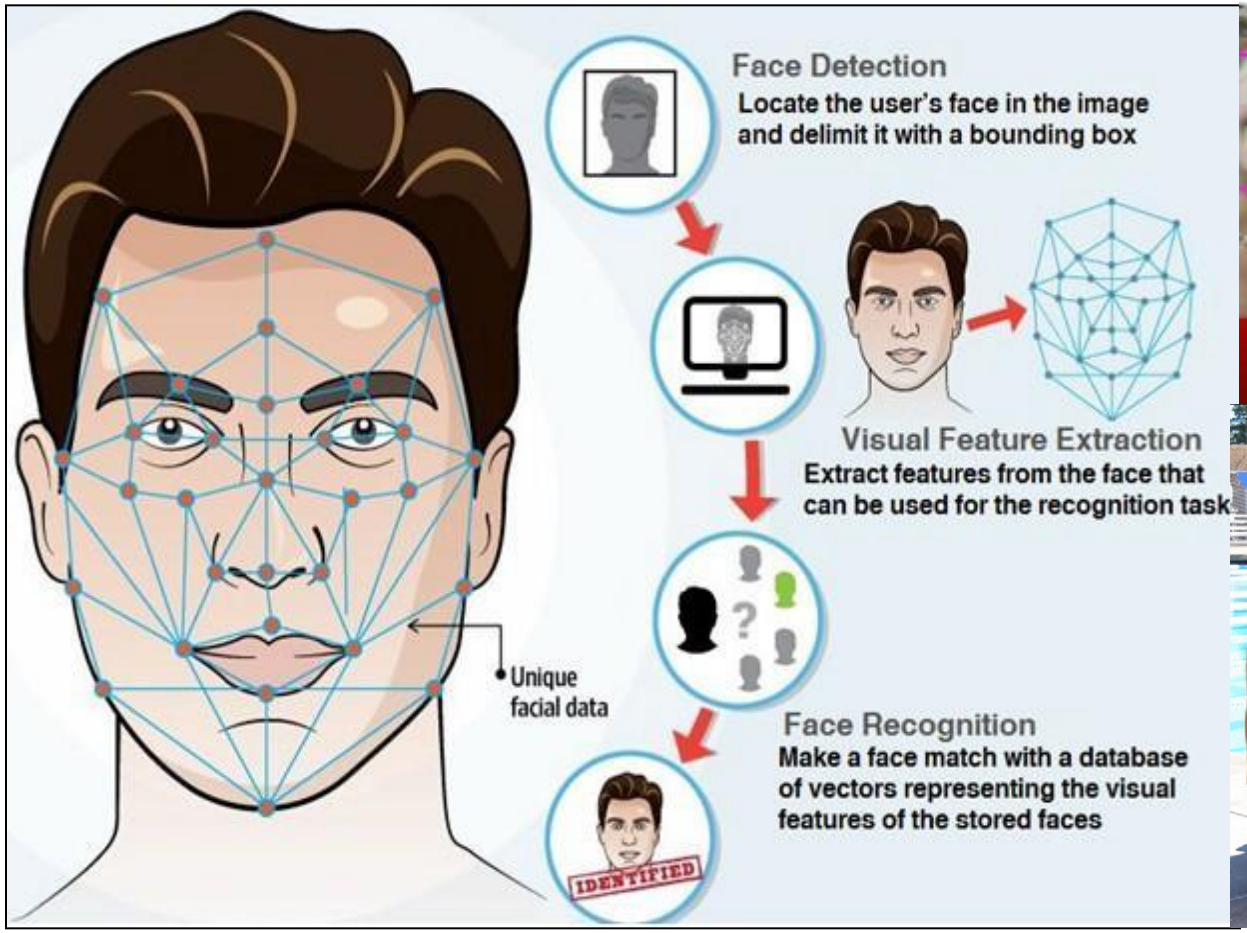
Over 80% of what people watch comes from our recommendations

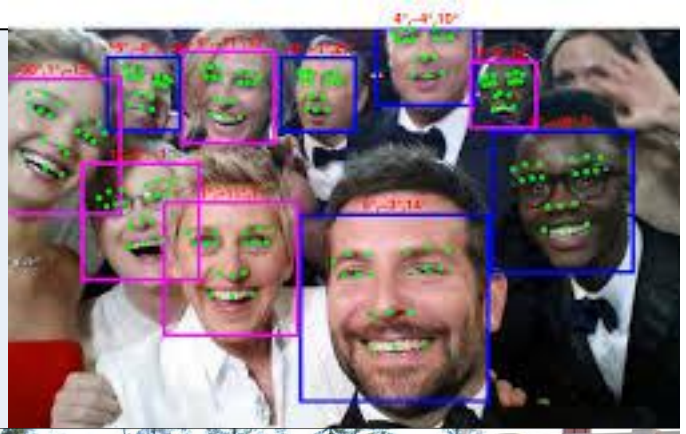
Recommendations are driven by **Machine Learning**



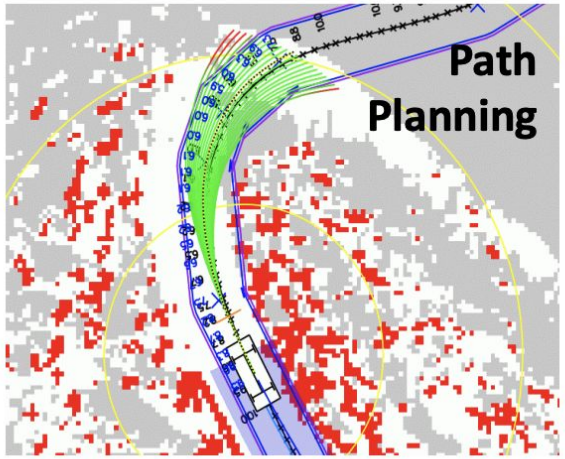
Netflix Challenge - $1 million prize to winner of 2009 contest - 100,480,507 ratings that 480,189 users gave to 17,770 movies. Winner was 10.1% better than the algorithm Netflix was using

ML & Face Recognition





ML for Autonomous Vehicles



ML Applications in Bioinformatics

- Used in Secondary Structure Prediction
- Used in Gene Finding
- Used in Sequence Motif Finding
- Used in SNP typing and GWAS Analysis
- Used in Disease Diagnosis
- Used in Nanopore DNA Sequencing
- Used in MS and NMR Spectra Prediction
- Used in Drug Design & Discovery
- Used in Protein Structure Prediction

Some ML Applications From My Lab

J. Listgarten *et al*. (2004) Clinical Cancer Res. 10: 2725-37.

Vol. 10, 2725–2737, April 15, 2004 Clinical Cancer Research 2725

Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms

Jennifer Listgarten,[4] Sambasivarao Damaraju,[1,4] Brett Poulin,[2] Lillian Cook,[4] Jennifer Dufour,[4] Adrian Driga,[4] John Mackey,[1,4] David Wishart,[3] Russ Greiner,[2] and Brent Zanke[1,4]

...ies of [1]Medicine, [2]Science, ...d the [4]Cross Cancer Institute of the ...onton, Alberta, Canada

...osition and causative environmental ...en recognized in human malignan-..., cancer cases occur sporadically, ...mental influences are critical in de-...o test the influence of genetic poly-...ancer risk, we have measured 98 ...rphisms (SNPs) distributed over 45 the genome are better at identifying breast cancer patients than any one SNP alone. As high-throughput technology for SNPs improves and as more SNPs are identified, it is likely that much higher predictive accuracy will be achieved and a useful clinical tool developed.

INTRODUCTION

Malignant transformation occurs through the accumulation of mutations in genes regulating cell division, apoptosis, invasiveness, or metastasis. These can occur as primary events or as a consequence of defects in "caretaker" genes that function in the maintenance of genomic stability (1). Inherited cancer predisposition from the inheritance of single genes almost exclusively results from abnormalities in DNA maintenance genes such as DNA double-strand break repair factors *BRCA1* or *BRCA2*, which are abnormal in familial breast cancer (2); the

REVIEW

Applications of Machine Learning in Cancer Prediction and Prognosis

Joseph A. Cruz, David S. Wishart

Departments of Biological Science and Computing Science, University of Alberta Edmonton, AB, Canada T6G 2E8

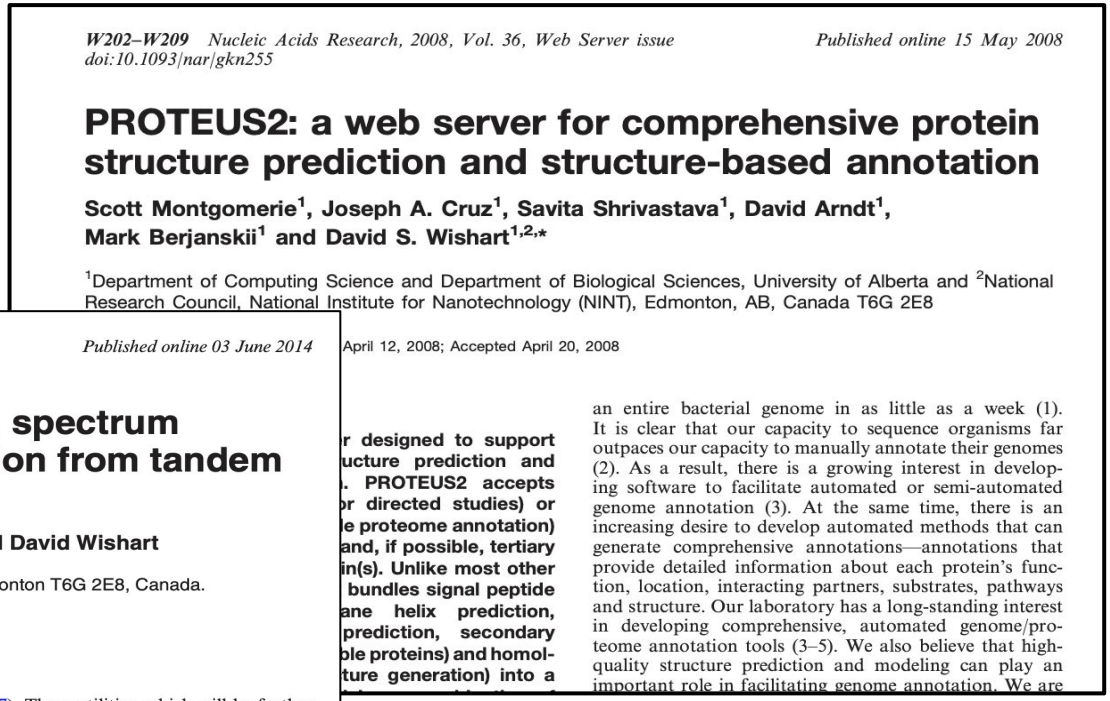
Abstract: Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to "learn" from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex proteomic and genomic measurements. As a result, machine learning is frequently used in cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. This latter approach is particularly interesting as it is part of a growing trend towards personalized, predictive medicine. In assembling this review we conducted a broad survey of the different types of machine learning methods being used, the types of data being integrated and the performance of these methods in cancer prediction and prognosis. A number of trends are noted, including a growing dependence on protein biomarkers and microarray data, a strong bias towards applications in prostate and breast cancer, and a heavy reliance on "older" technologies such artificial neural networks (ANNs) instead of more recently developed or more easily interpretable machine learning methods. A number of published studies also appear to lack an appropriate level of validation or testing. Among the better designed and validated studies it is clear that machine learning methods can be used to substantially (15-25%) improve the accuracy of predicting cancer susceptibility, recurrence and mortality. At a more fundamental level, it is also evident that machine learning is also helping to improve our basic understanding of cancer development and progression.

Keywords: Cancer, machine learning, prognosis, risk, prediction

J. A. Cruz *et al*. (2006) Cancer Informatics. 2: 59-77.

Some ML Applications From My Lab

S. Montgomerie *et al*. (2008) Nucl. Acids Res. 36: W202-9.



W202–W209 *Nucleic Acids Research, 2008, Vol. 36, Web Server issue* *Published online 15 May 2008*
doi:10.1093/nar/gkn255

PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation

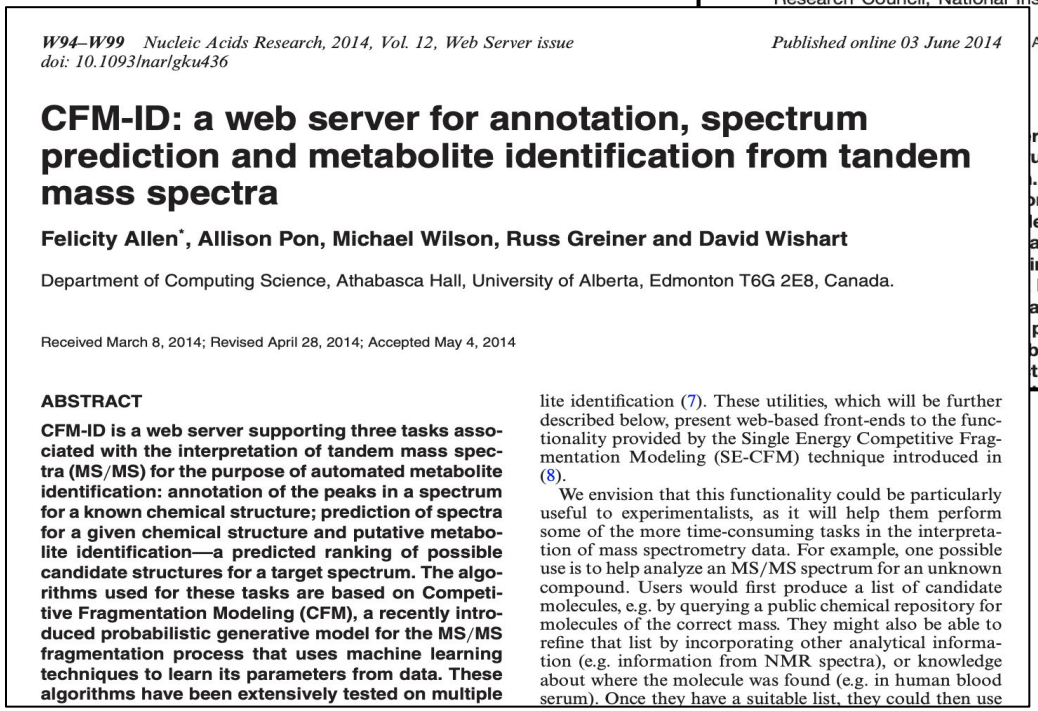
Scott Montgomerie[1], Joseph A. Cruz[1], Savita Shrivastava[1], David Arndt[1], Mark Berjanskii[1] and David S. Wishart[1,2,*]

[1]Department of Computing Science and Department of Biological Sciences, University of Alberta and [2]National Research Council, National Institute for Nanotechnology (NINT), Edmonton, AB, Canada T6G 2E8

April 12, 2008; Accepted April 20, 2008

r designed to support
ucture prediction and
. PROTEUS2 accepts
r directed studies) or
e proteome annotation)
and, if possible, tertiary
in(s). Unlike most other
bundles signal peptide
ane helix prediction,
prediction, secondary
ble proteins) and homol-
ture generation) into a

an entire bacterial genome in as little as a week (1). It is clear that our capacity to sequence organisms far outpaces our capacity to manually annotate their genomes (2). As a result, there is a growing interest in developing software to facilitate automated or semi-automated genome annotation (3). At the same time, there is an increasing desire to develop automated methods that can generate comprehensive annotations—annotations that provide detailed information about each protein's function, location, interacting partners, substrates, pathways and structure. Our laboratory has a long-standing interest in developing comprehensive, automated genome/proteome annotation tools (3–5). We also believe that high-quality structure prediction and modeling can play an important role in facilitating genome annotation. We are



W94–W99 *Nucleic Acids Research, 2014, Vol. 12, Web Server issue* *Published online 03 June 2014*
doi: 10.1093/nar/gku436

CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra

Felicity Allen*, Allison Pon, Michael Wilson, Russ Greiner and David Wishart

Department of Computing Science, Athabasca Hall, University of Alberta, Edmonton T6G 2E8, Canada.

Received March 8, 2014; Revised April 28, 2014; Accepted May 4, 2014

ABSTRACT

CFM-ID is a web server supporting three tasks associated with the interpretation of tandem mass spectra (MS/MS) for the purpose of automated metabolite identification: annotation of the peaks in a spectrum for a known chemical structure; prediction of spectra for a given chemical structure and putative metabolite identification—a predicted ranking of possible candidate structures for a target spectrum. The algorithms used for these tasks are based on Competitive Fragmentation Modeling (CFM), a recently introduced probabilistic generative model for the MS/MS fragmentation process that uses machine learning techniques to learn its parameters from data. These algorithms have been extensively tested on multiple

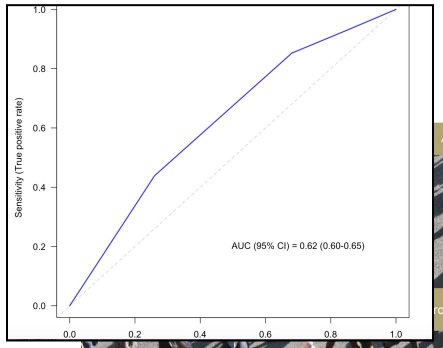
lite identification (7). These utilities, which will be further described below, present web-based front-ends to the functionality provided by the Single Energy Competitive Fragmentation Modeling (SE-CFM) technique introduced in (8).

We envision that this functionality could be particularly useful to experimentalists, as it will help them perform some of the more time-consuming tasks in the interpretation of mass spectrometry data. For example, one possible use is to help analyze an MS/MS spectrum for an unknown compound. Users would first produce a list of candidate molecules, e.g. by querying a public chemical repository for molecules of the correct mass. They might also be able to refine that list by incorporating other analytical information (e.g. information from NMR spectra), or knowledge about where the molecule was found (e.g. in human blood serum). Once they have a suitable list, they could then use

F. Allen *et al*. (2014) Nucleic Acids Res. 42: W94-99.

ML Applications to GWAS

- Calculating optimal SNP panels from GWAS studies for biomarker determination
- Using SVM or Random Forest regression to calculate multi-SNP ROC curves

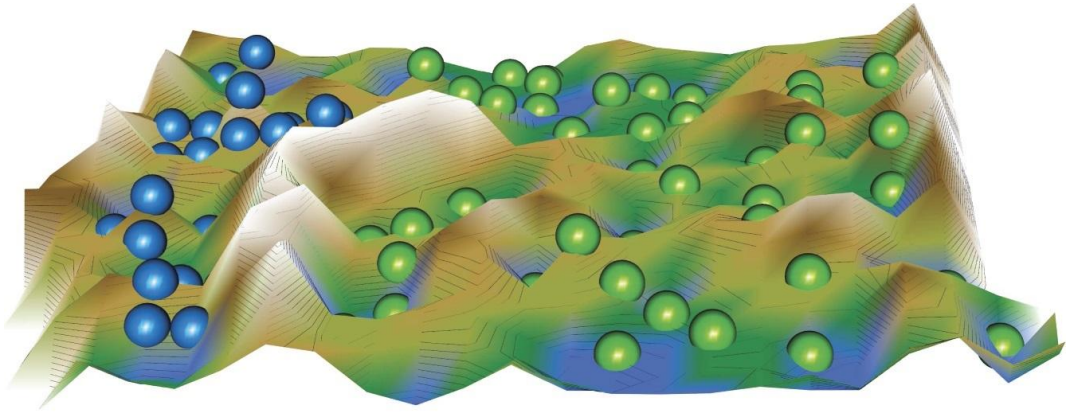


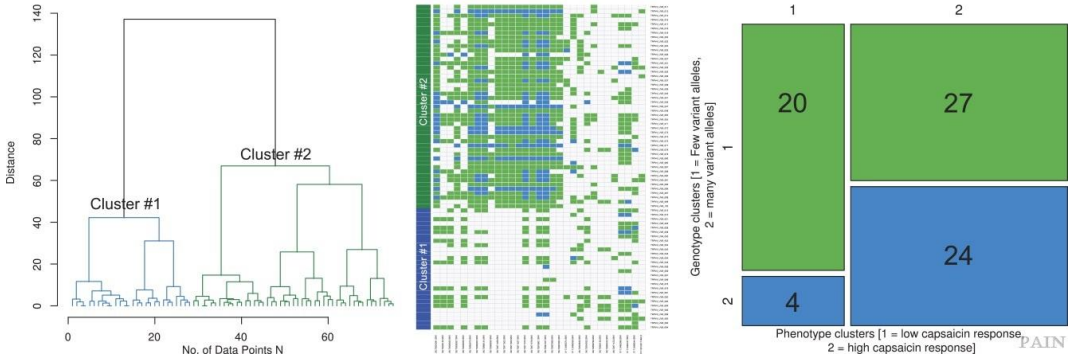
J. Patron et al. PLoS One. 2019 Dec 5;14(12):e0220215.

www.gwasrocs.ca

ML Applications to SNP Typing

- **Associating multiloci genotypes or SNP variants (100s to 1000s) with categorical phenotypes**



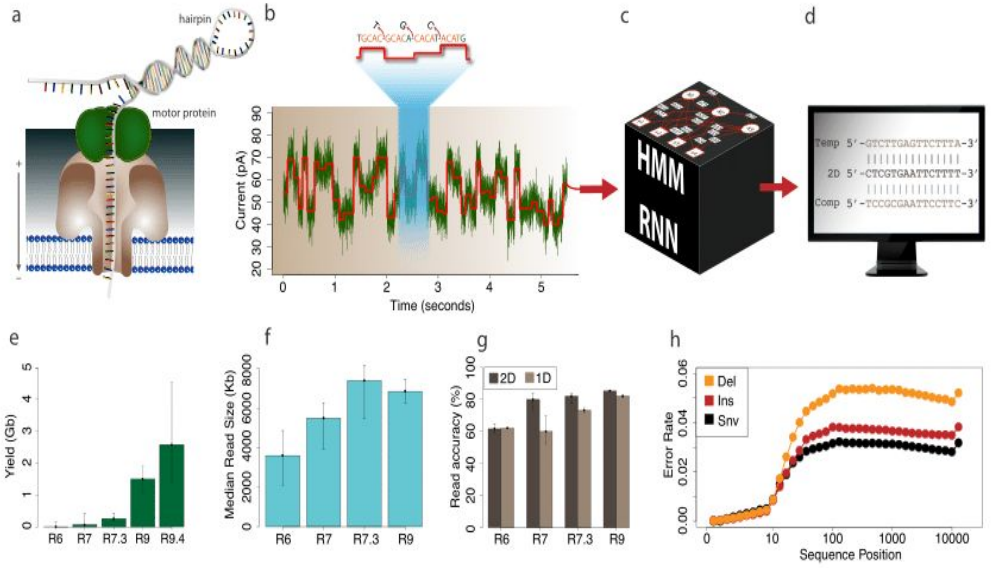


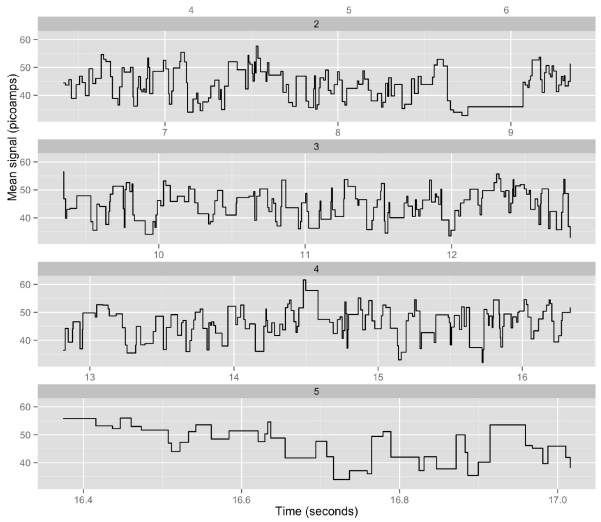
D. Kringel *et al*. Machine-learned analysis of the association of NGS based TRPV1 and TRPA1 genotypes with sensitivity to heat stimuli. Pain, 2018 July 159(7): 1366-81

75 patients - 24 sensitive, 51 not-sensitive patients, 278 initial loci, reduced to 31 gene loci. Used non-supervised swarm clustering (above)

ML Applications in DNA Sequencing

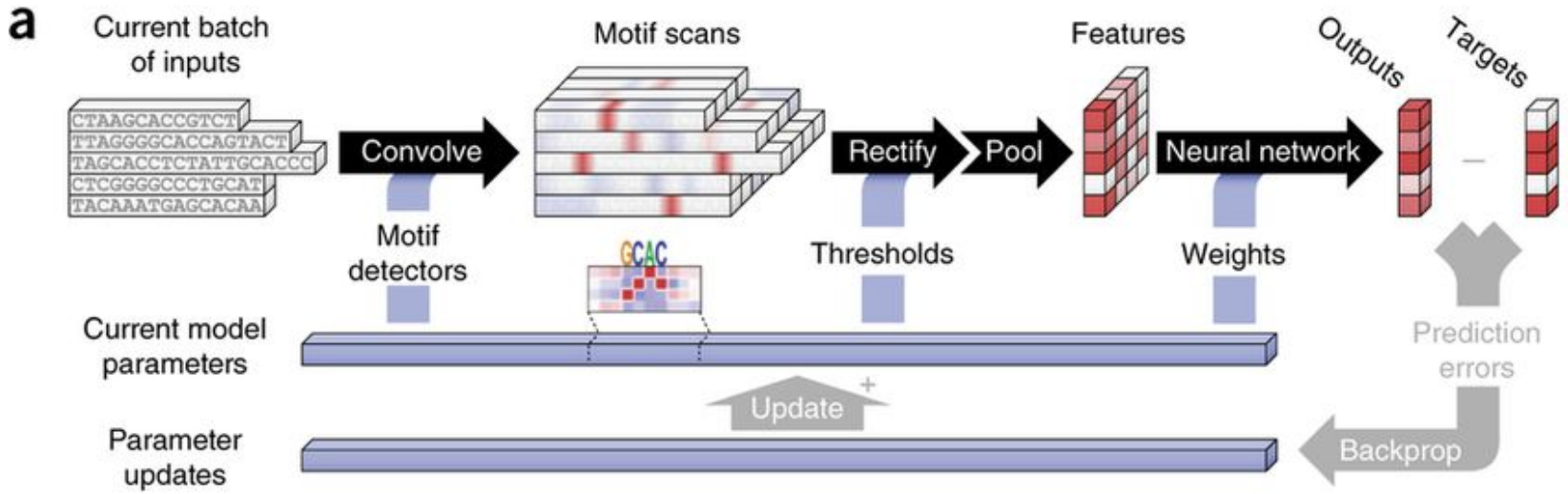
- **Reading DNA sequence data for the Oxford Nanopore's MinION DNA sequencer**





ML Applications in Bioinformatics

- **DeepBind – a DNN tool for predicting the sequence specificities of DNA and RNA binding proteins (B. Alipanahi *et al*., (2015) Nature Biotech. 33:831)**
- **Can be trained on *in vitro* data yet works well on *in vivo* predictions**



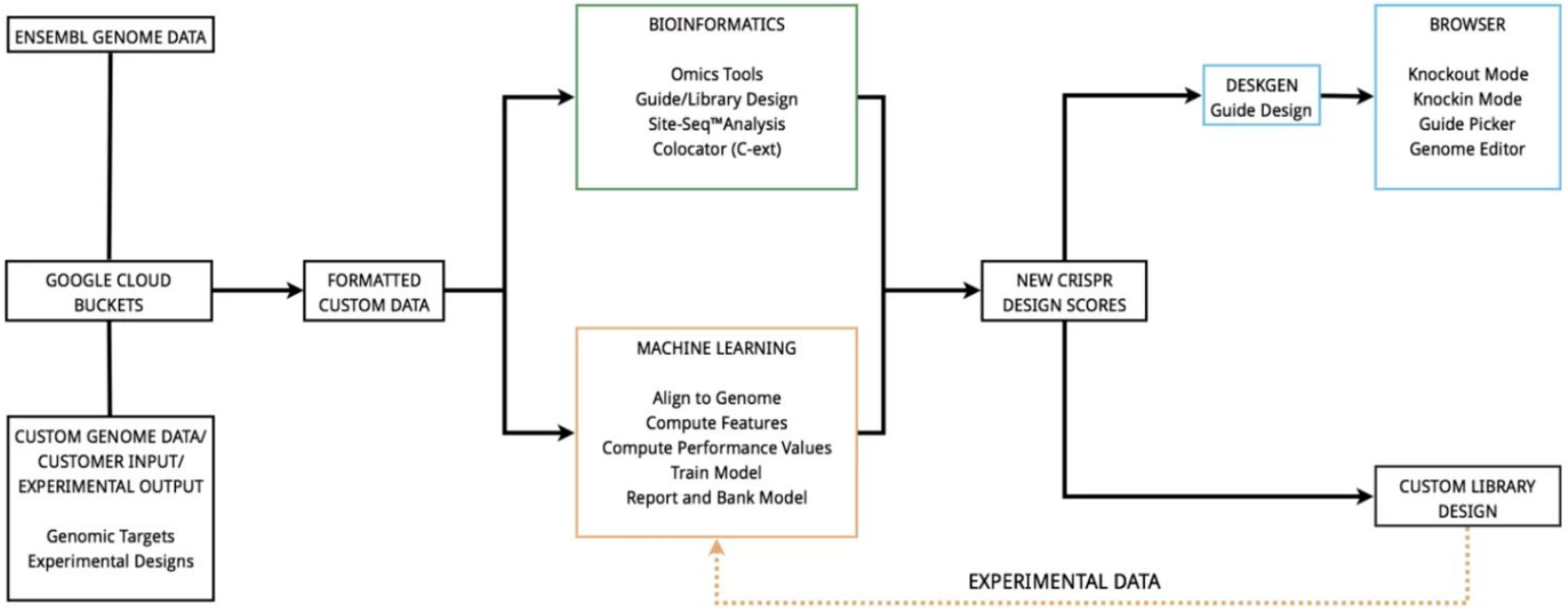
ML Applications in Genomics

- DeepBioSeq – a new tool that uses deep learning (convolution neural networks) to analyze RNA-Seq data
- Does not require sequence preprocessing or genomic alignment
- Works directly with .fastq files
- Uses quality of the raw reads to help
- Can be adapted to single cell sequencing and ChIP sequence data

https://www.sbi.uni-rostock.de/research/projects/detail/51

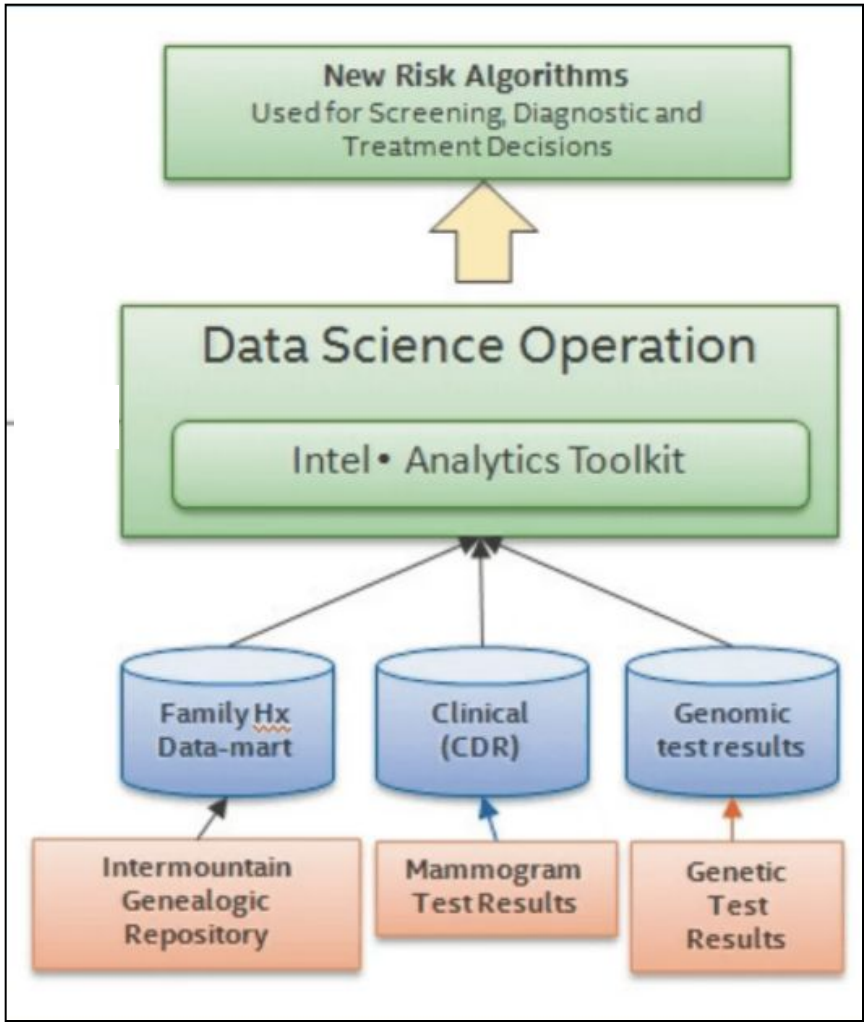
ML Applications in Molecular Biology

- **Optimizing the design of CRISPR target sequences (Desktop Genetics, UK)**



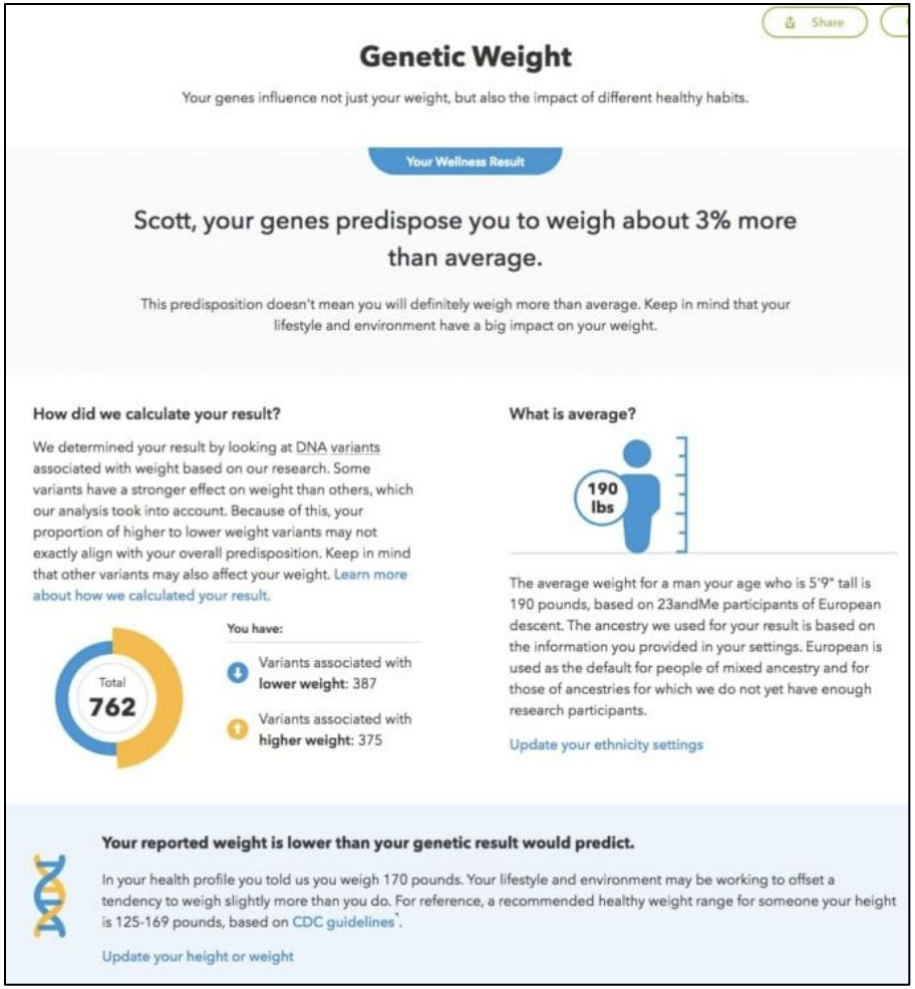
ML Applications in Cancer

- Clinical system support tool for cancer screening at Intermountain Health in Salt Lake City Utah
- Centralized DB of genomic data linked to clinical and patient history data
- Linked to EHR data & genealogical data
- ML helps to integrate & interpret combined data sets
- Found to accelerate and improve cancer risk assessment



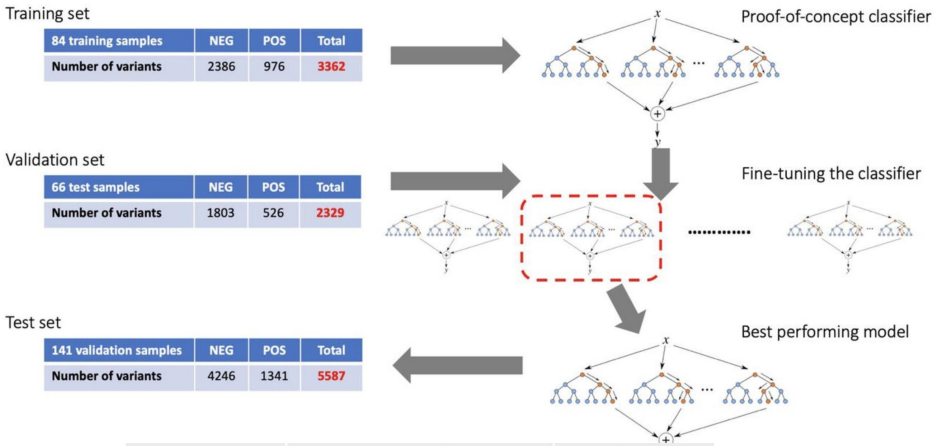
ML Applications in Genomics

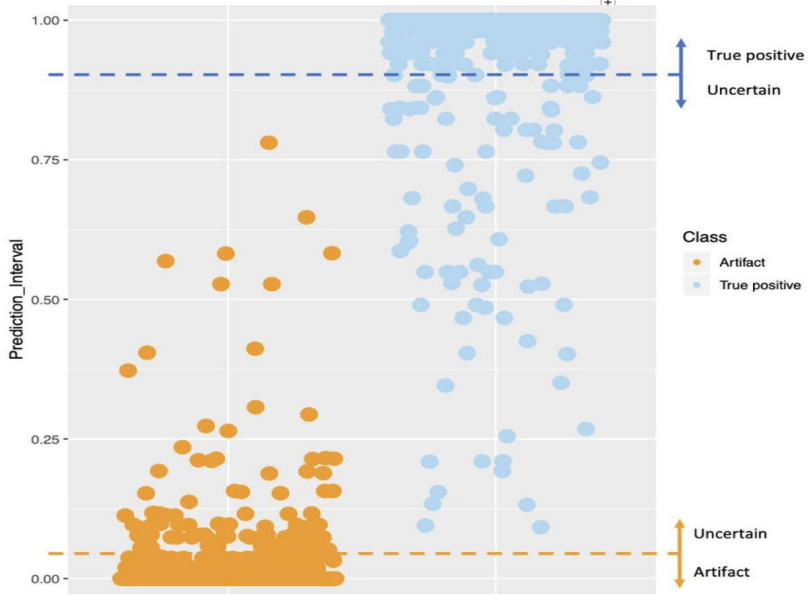
- 23andMe's genetic weight report
- Used machine learning data on 600,000 people to combine GWAS data on BMI with lifestyle data to generate a model for weight prediction
- Allows creation of a personalized weight report for 23&Me users



ML Application in Tumor Genomics

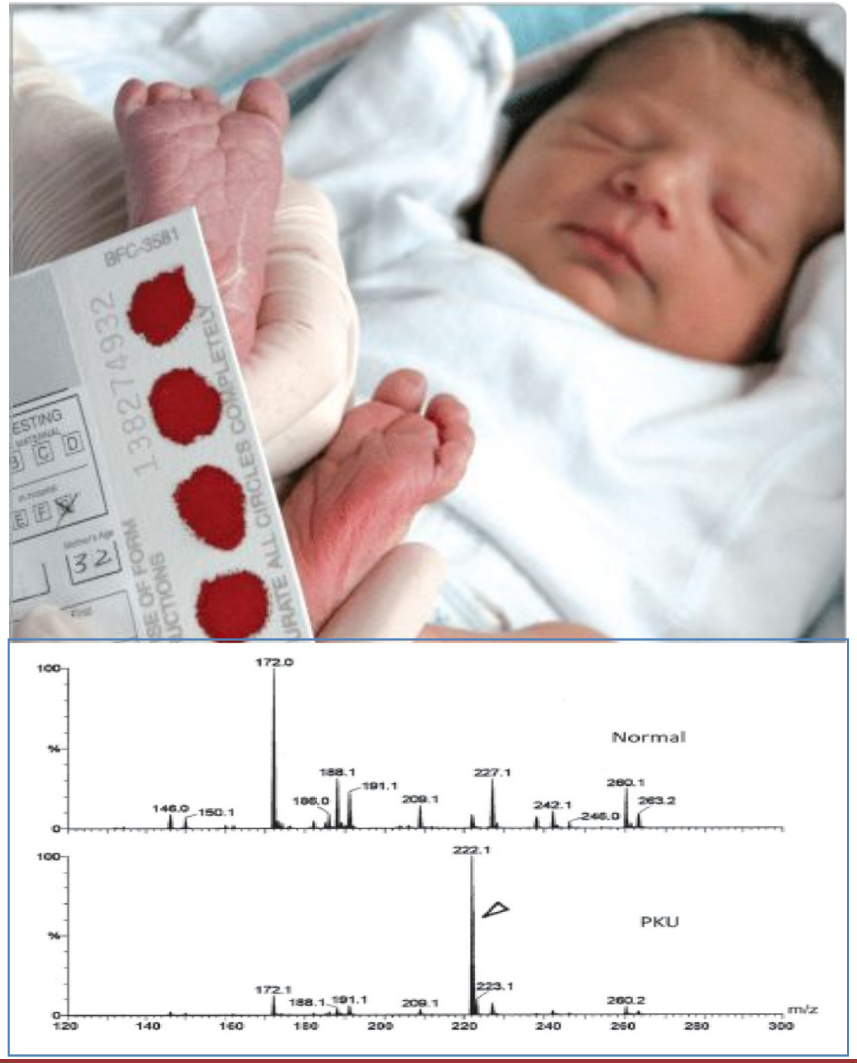
- Using ML to call single nucleotide variants (SNVs) in tumor samples sequenced via NGS (PMID:31672855)
- Uses Random Forest and features including
  - Number of unique reads supporting alternate allele
  - Strand bias
  - Variant allele fraction
  - Dissimilarity to normals
  - Batch effect metrics
- Achieved 100% specificity and 97% sensitivity – better than humans



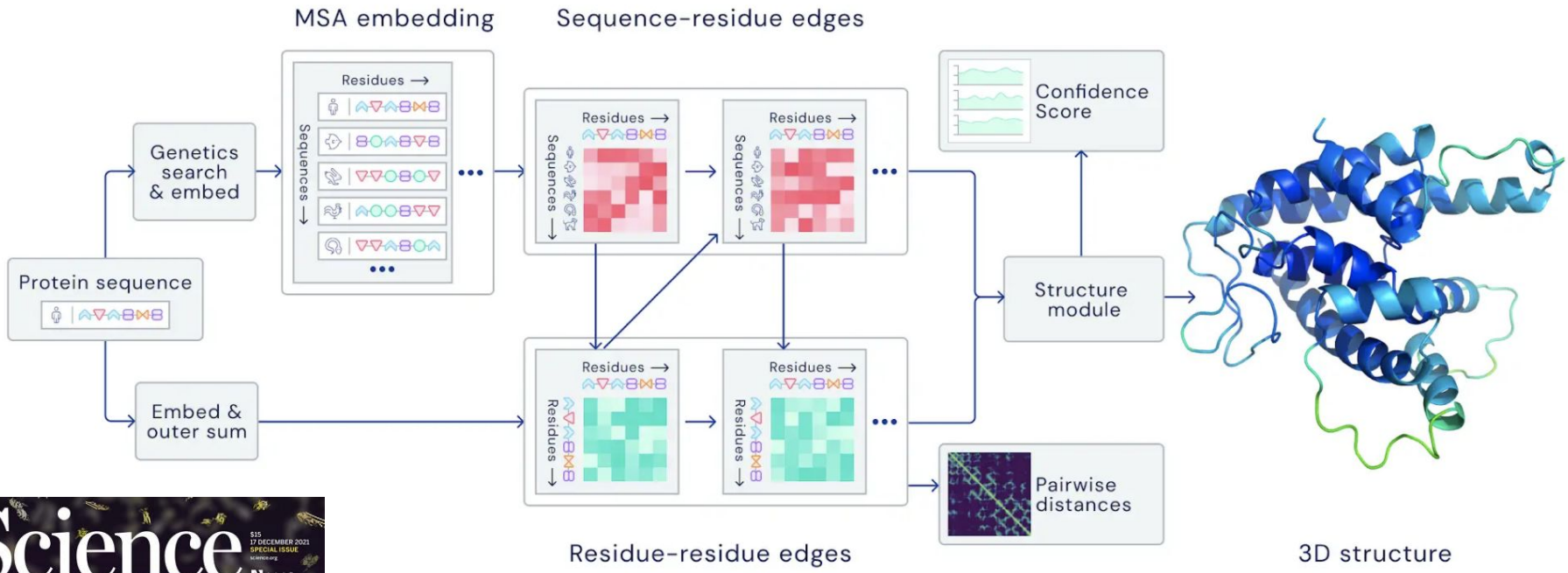


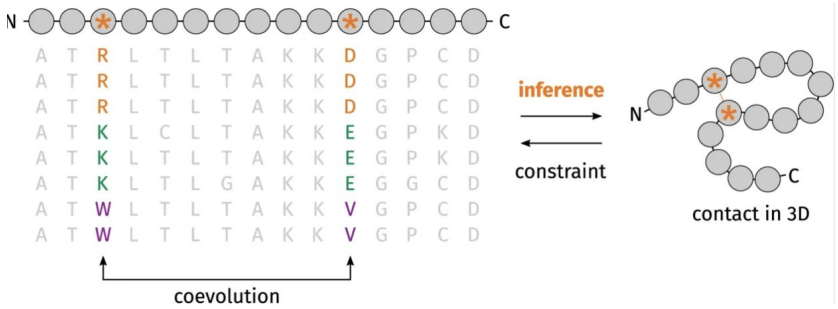
ML Applications in Newborn Screening

- Newborn screening aided by machine learning (PMID:23702487)
- Number of false positives were reduced from 21 to 2 for PKU, 30 to 10 for hypermethioninemia, and 209 to 46 for 3-MCC deficiency

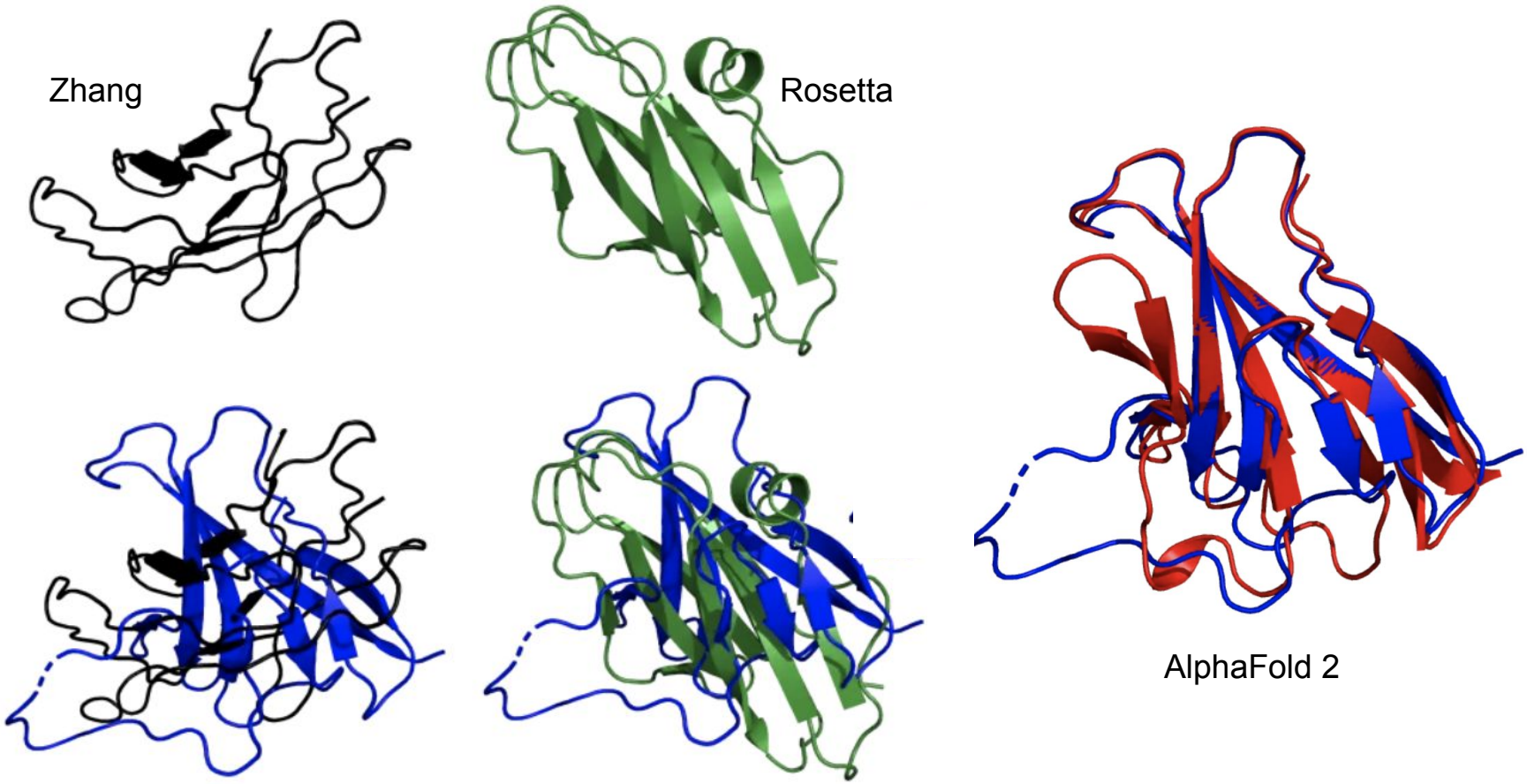


ML Applications in Structure Biology (AlphaFold)

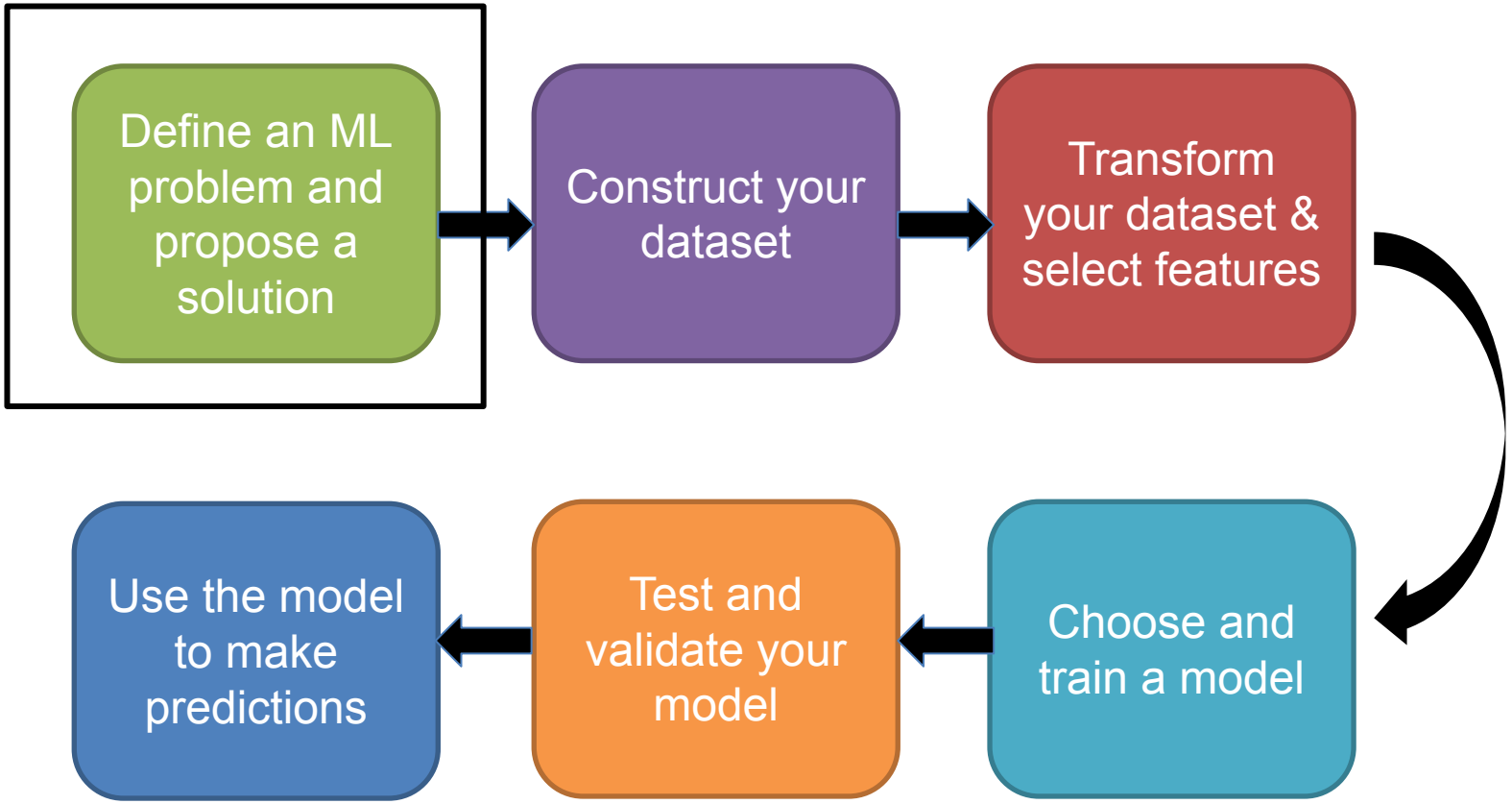




AlphaFold2 vs. Rosetta & Zhang for COVID Orf8



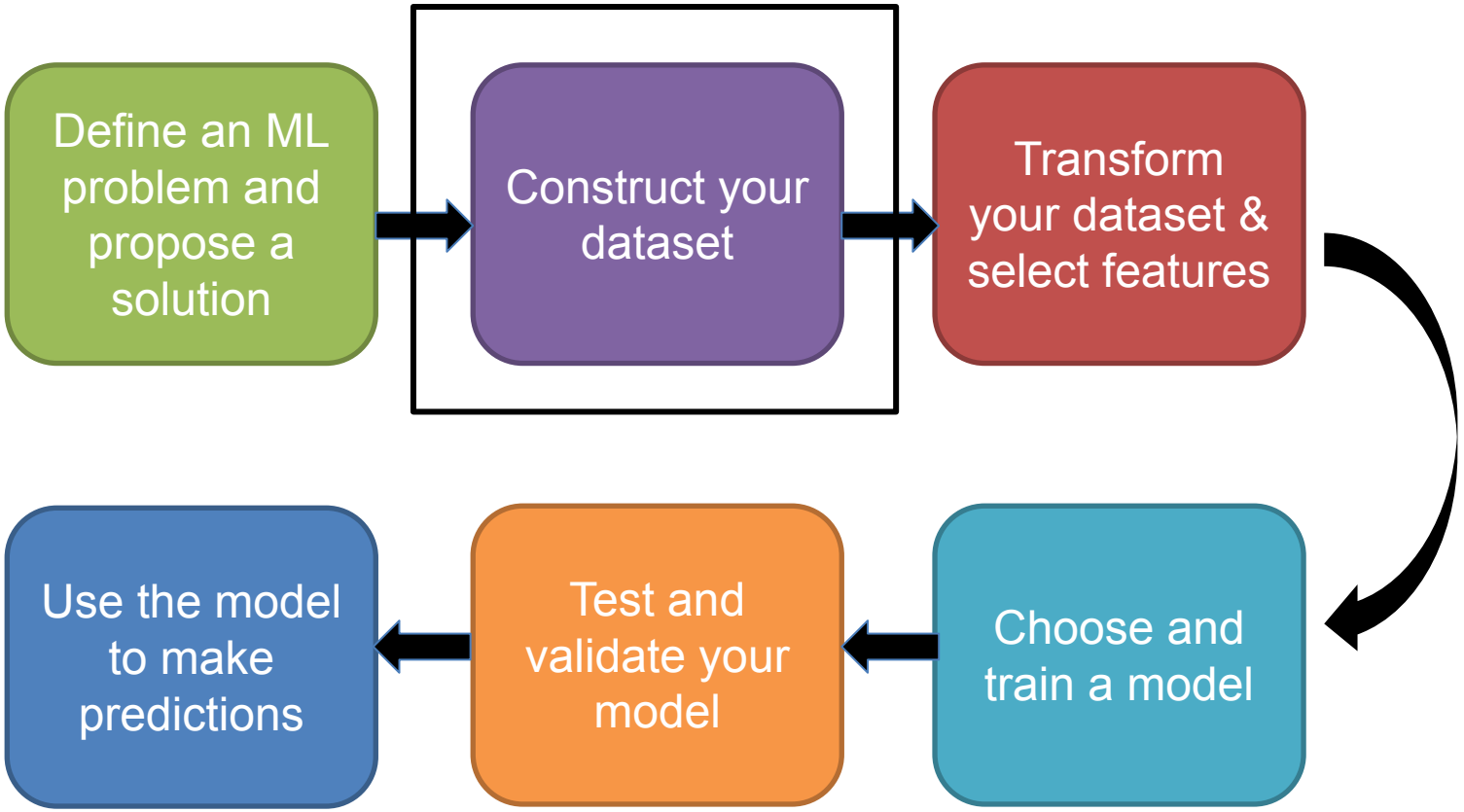
Machine Learning Workflow



Choose Your Problem

- Choose an “unsolved” problem that is interesting to you or your research
- The problem should be something that can’t be (easily) solved mathematically or something which is very difficult to do or requires special knowledge/training
- Ideally it should be focused on finding a pattern, a set of features or a way of classifying groups
- You also need to choose a problem with lots of training or exemplar data where the answer is known or mostly known

Machine Learning Workflow



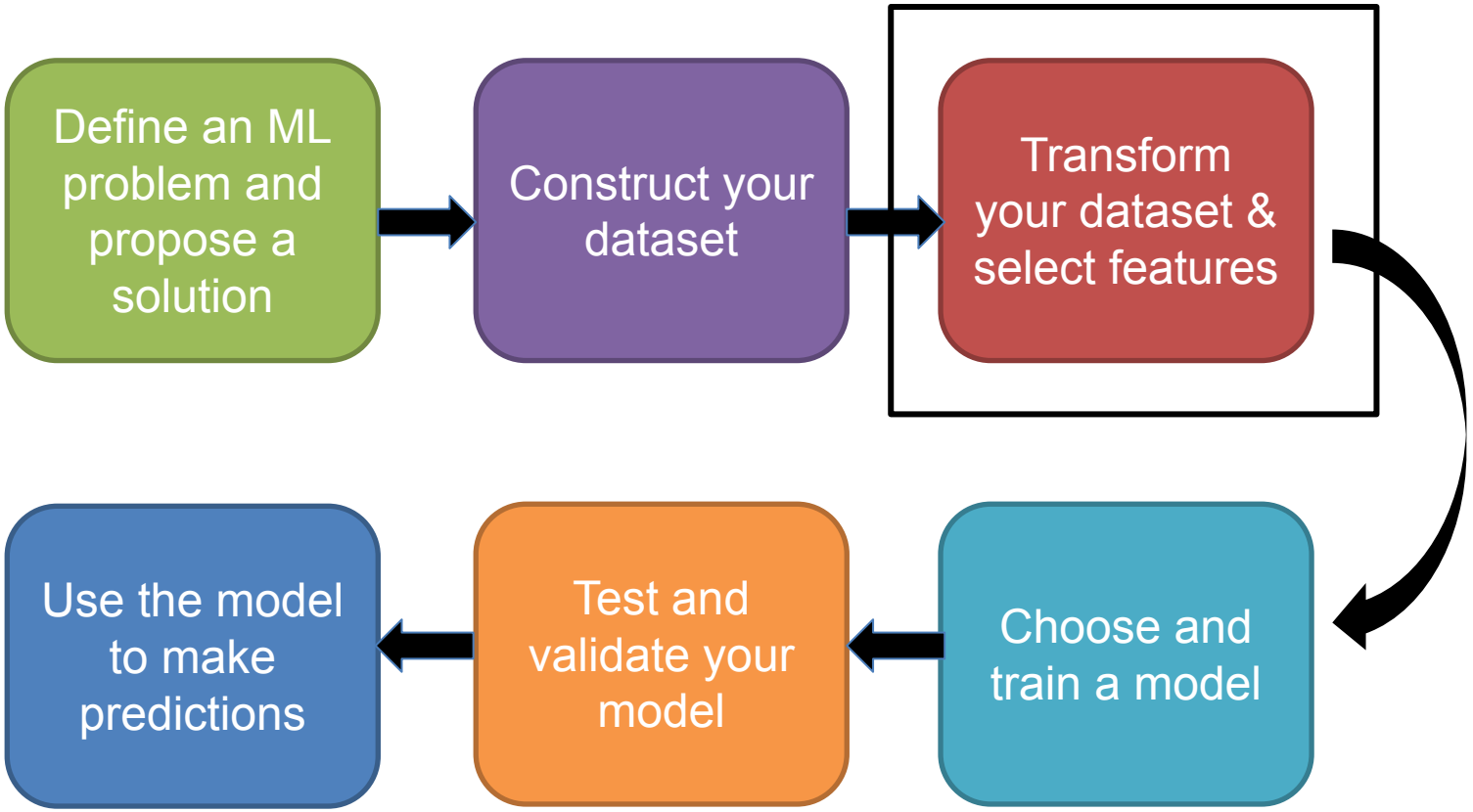
Construct Your Dataset

- Must obtain your data from reliable sources
- Needs to be labeled data (categorical, nominal or numerical) where the labels are gold-standard “answers”
- Needs to have relevant parameters that you think or you know contribute to the phenomenon (phase of moon or astrological signs probably don’t help in predicting DNA binding motifs)

Construct Your Dataset

- ML requires: 1) training data; 2) testing data and 3) validation data
- No single “right” answer to amount of data required for machine learning
- At a minimum, collect about 1000 examples
- For most "average" problems, you need to have 10,000 - 100,000 examples
- For “hard” problems like machine translation, high dimensional data generation, or anything requiring deep learning, you should try to get 100,000 - 1,000,000 examples

Machine Learning Workflow



Data Transformation

- Clean up the data (remove repeats, fill in or impute missing values, reformat for compatibility, fix outliers, group sparse classes, etc.) – data cleansing
- Convert categorical or nominal data to numeric data, one-hot encoding, normalize skewed data, range scale – data transformation and feature engineering
- Add features, add obvious relationships, select features, keep relevant data remove irrelevant data (intuition or statistics) – feature selection

One-hot Encoding

- **Method of converting categorical or nominal variables into a binary representation – better for ML algorithms**

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Encoding vs. Embedding

- One-hot encoding is fast and simple, but it loses context and information about surrounding characters or data
- Embedding takes context into account and gives features with similar influence in a data string (sequences, temporal data) similar value for a specific feature (but it is slow)
- Embedding is very useful in NLP such as named entity recognition, text summarization, co-reference resolution as well as bioinformatics applications such as gene finding and protein structure prediction

Encoding vs. Embedding

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

	Femininity	Youth	Royalty
Man	0	0	0
Woman	1	0	0
Boy	0	1	0
Girl	1	1	0
Prince	0	1	1
Princess	1	1	1
Queen	1	0	1
King	0	0	1
Monarch	0.5	0.5	1

Easy to code
Very sparse matrix
Large amount of data
No context

Harder to code
Denser data
Low dimension matrix
Keeps context

Feature Engineering

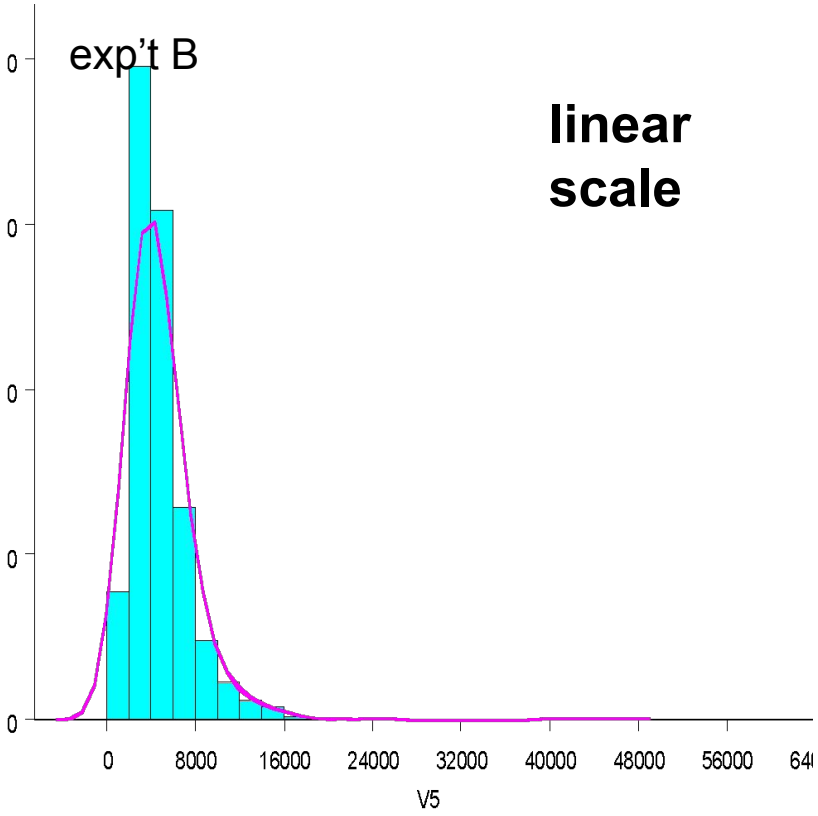
- Feature engineering involves scaling, transforming, normalizing, or standardizing
- It involves transforming the data to make it more suitable for modeling
- Helps improve model performance, reduces the impact of outliers, and ensures that the data is on the same scale
- Feature scaling transforms the values of variables in a dataset to a similar scale
- Necessary when variables have different ranges, units of measurement, or orders of magnitude

Scaling vs. Normalizing vs. Standardizing

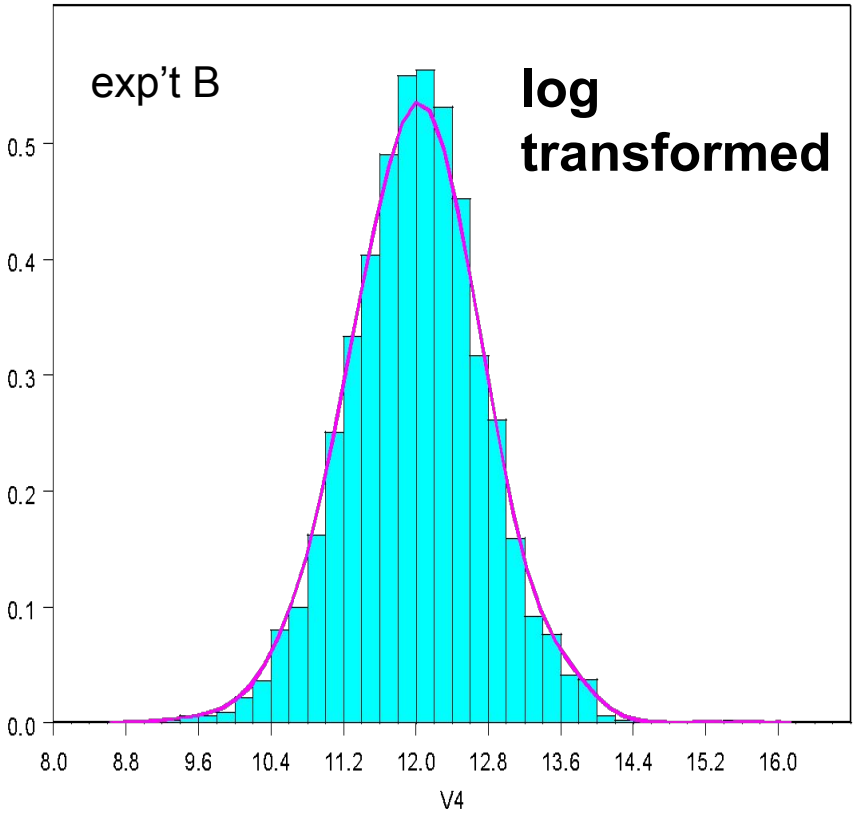
- Scaling is important for ML algorithms that use derivatives for optimization (logistic regression, PLS-DA, ANNs, SVMs, KNNs, etc.)
- Scaling facilitates meaningful comparisons between features, improves model convergence, stops certain features from overwhelming others
- Normalizing is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1
- Standardizing is a scaling method where the values are centered around the mean with SD=1

Fixing Skewed Data with a Log Transformation

Skewed distribution



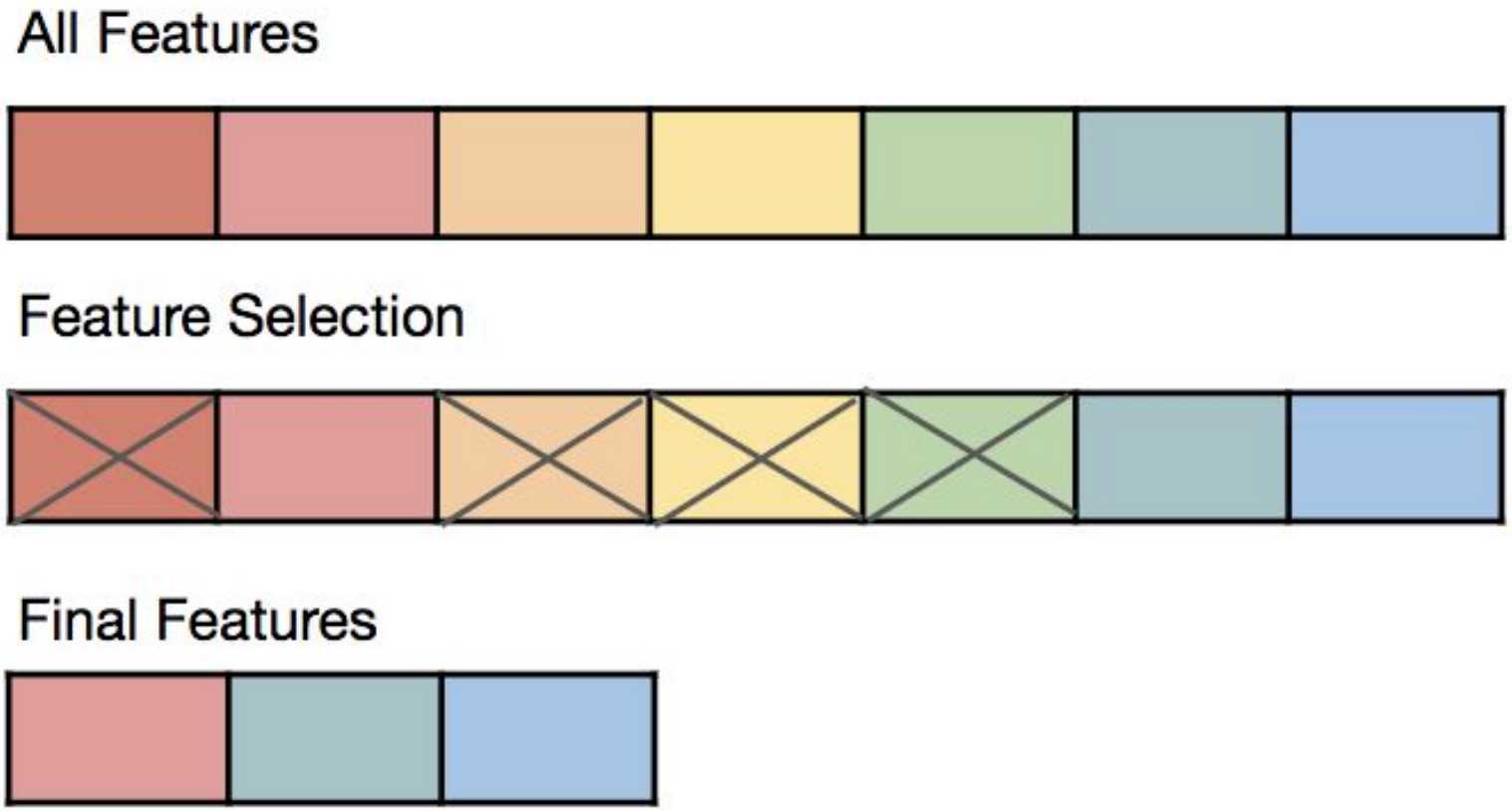
Normal distribution



Feature Selection

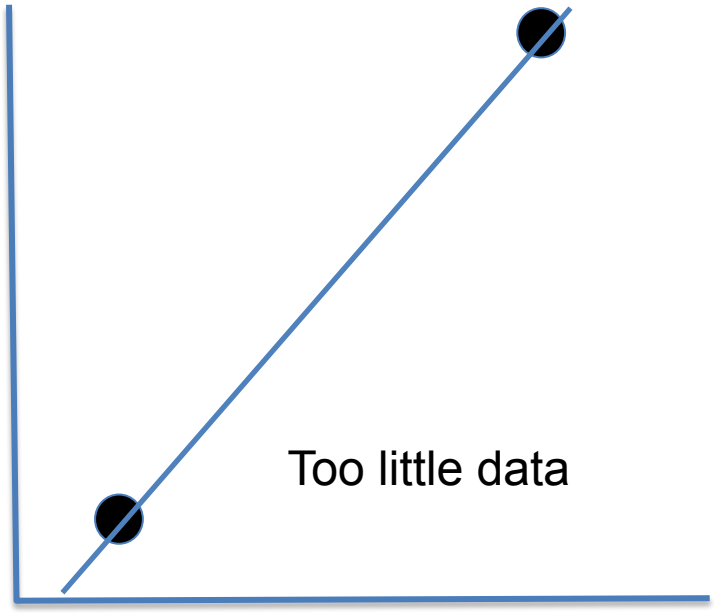
- Not all of the data collected for training a machine learning model is relevant
- Feature selection is a way of reducing the amount of data used in training an ML model
- Feature selection is a way to automatically or manually select data features that contribute most to the prediction accuracy of the model
- Including irrelevant features can reduce the accuracy of ML models

Feature Selection

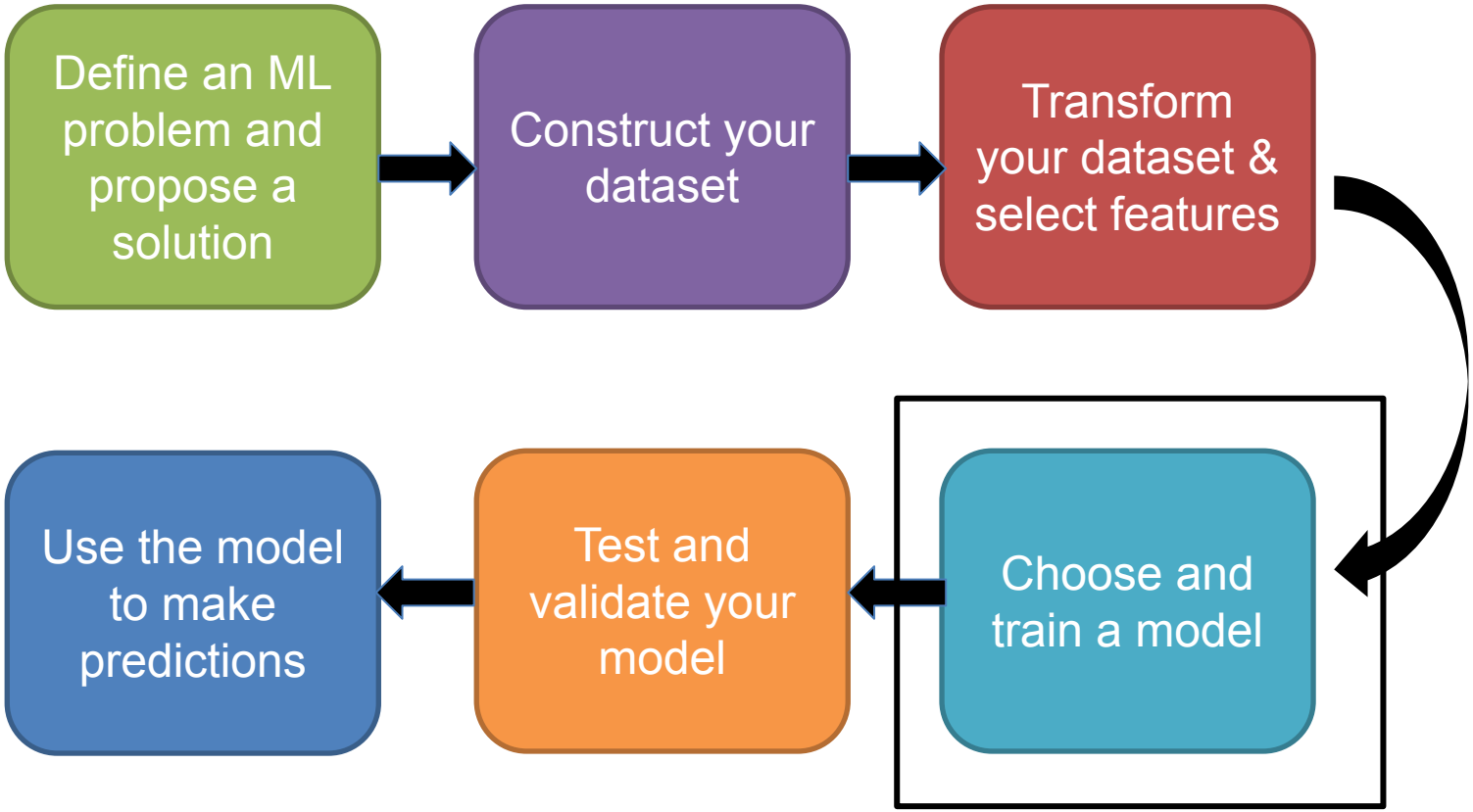


Feature Selection

- **Sometimes you can train your model on too little data or too few data points**



Machine Learning Workflow

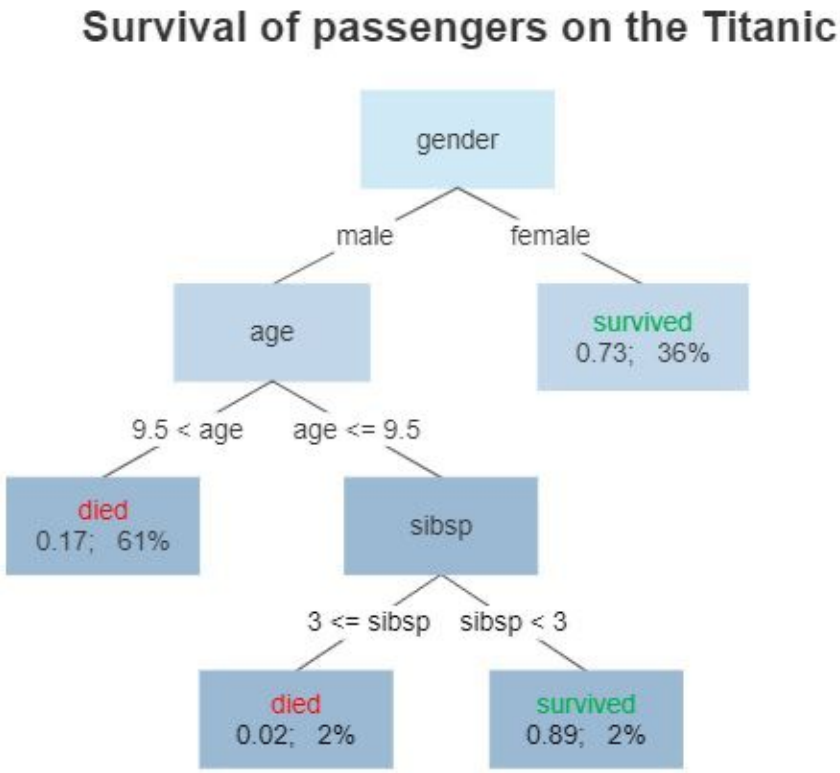


ML Uses Different "Models"

- All Machine Learning approaches require the use or creation of certain kinds of models
- Common models include:
  - Decision Trees/Random Forest
  - Artificial Neural Networks (ANNs)
  - Hidden Markov Models (HMMs)
  - Support Vector Machines (SVMs)
  - Genetic Algorithms (GAs)
  - Bayesian Networks (BNs)
- No *a priori* way of know which model is best – often you have to try many

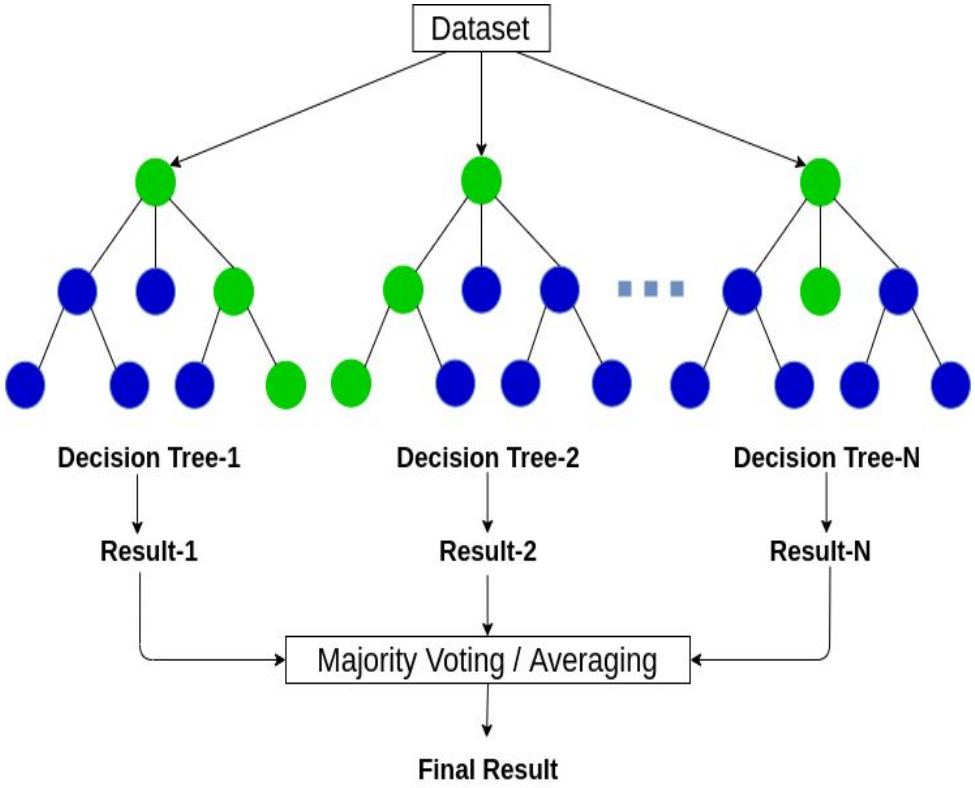
Decision Tree

- Simplest ML algorithm to understand and implement
- Computer learns to split, categorize or regress data based on decisions (greater than, less than, yes/no) and “cost” of decisions
- Tree-like structure consists of branches (edges) and leaves (nodes)
- No need for data scaling



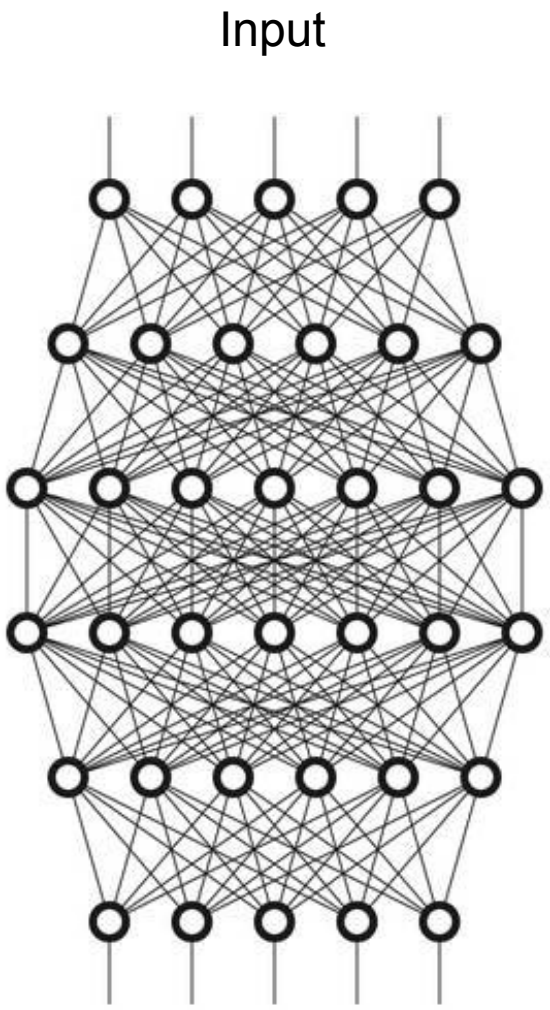
Random Forest

- Combines many decision trees into a “forest”
- Known as an ensemble method for learning
- Can be used for both classification and regression
- Tries to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree



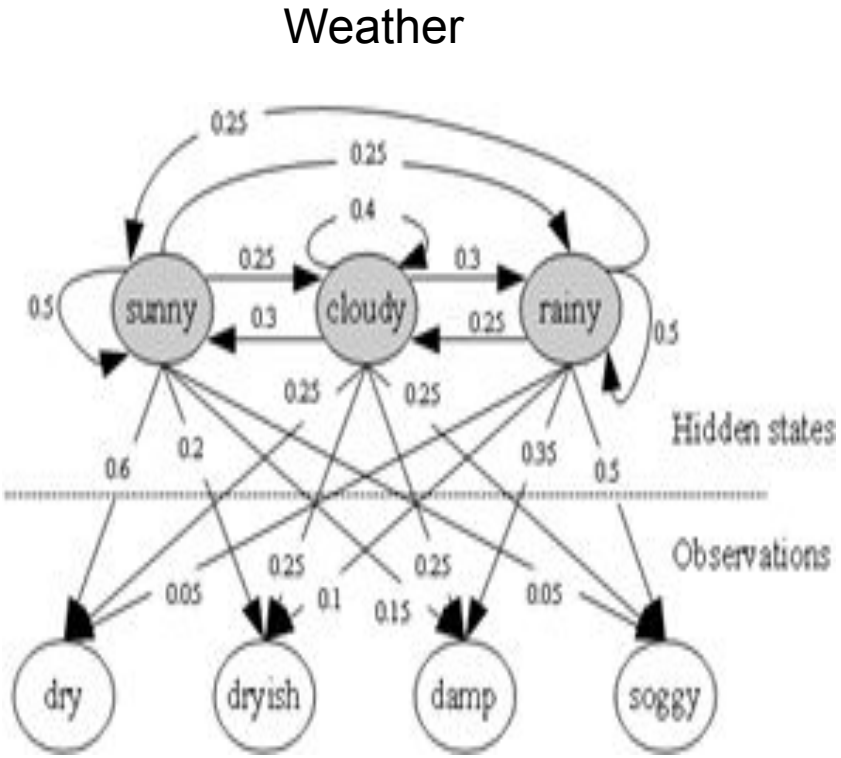
Artificial Neural Network

- Attempts to simulate the function and activity of the brain using inputs, outputs and neurons
- ANNs are based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain
- Can be used for both classification and regression



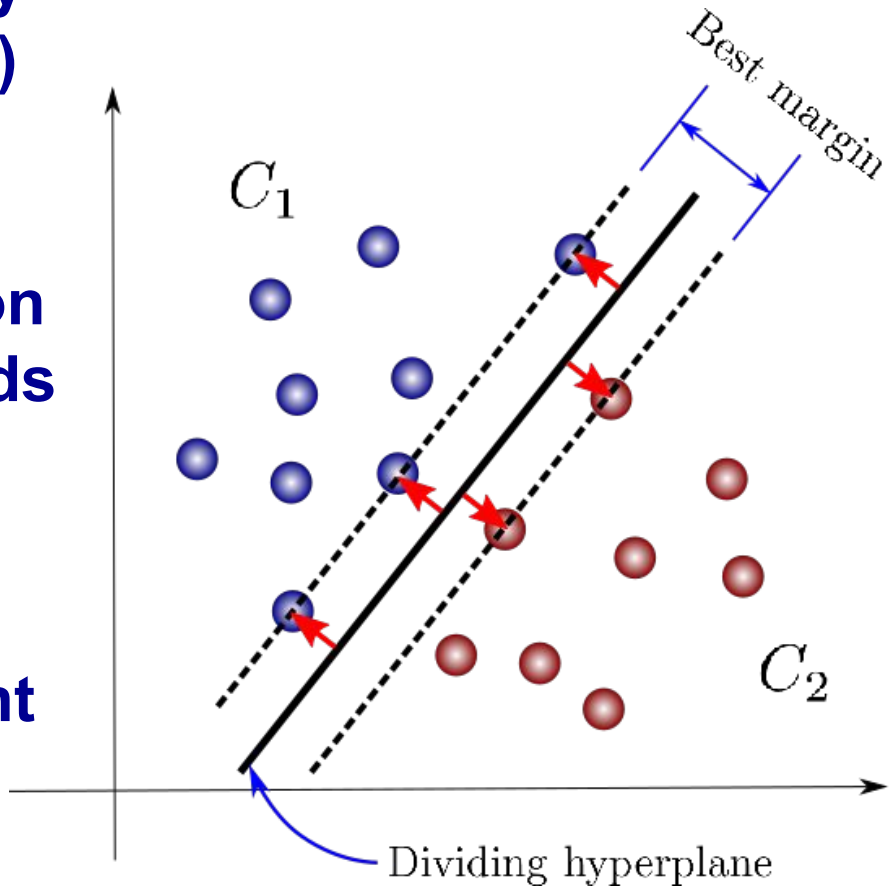
Hidden Markov Model

- HMMs are probabilistic graphical models designed to model a sequence as a result of a Markovian process that cannot be observed
- Consist of emission and transition probabilities that must be optimized via dynamic programming
- Useful for predicting time trends, sequential events or interpreting sequence data

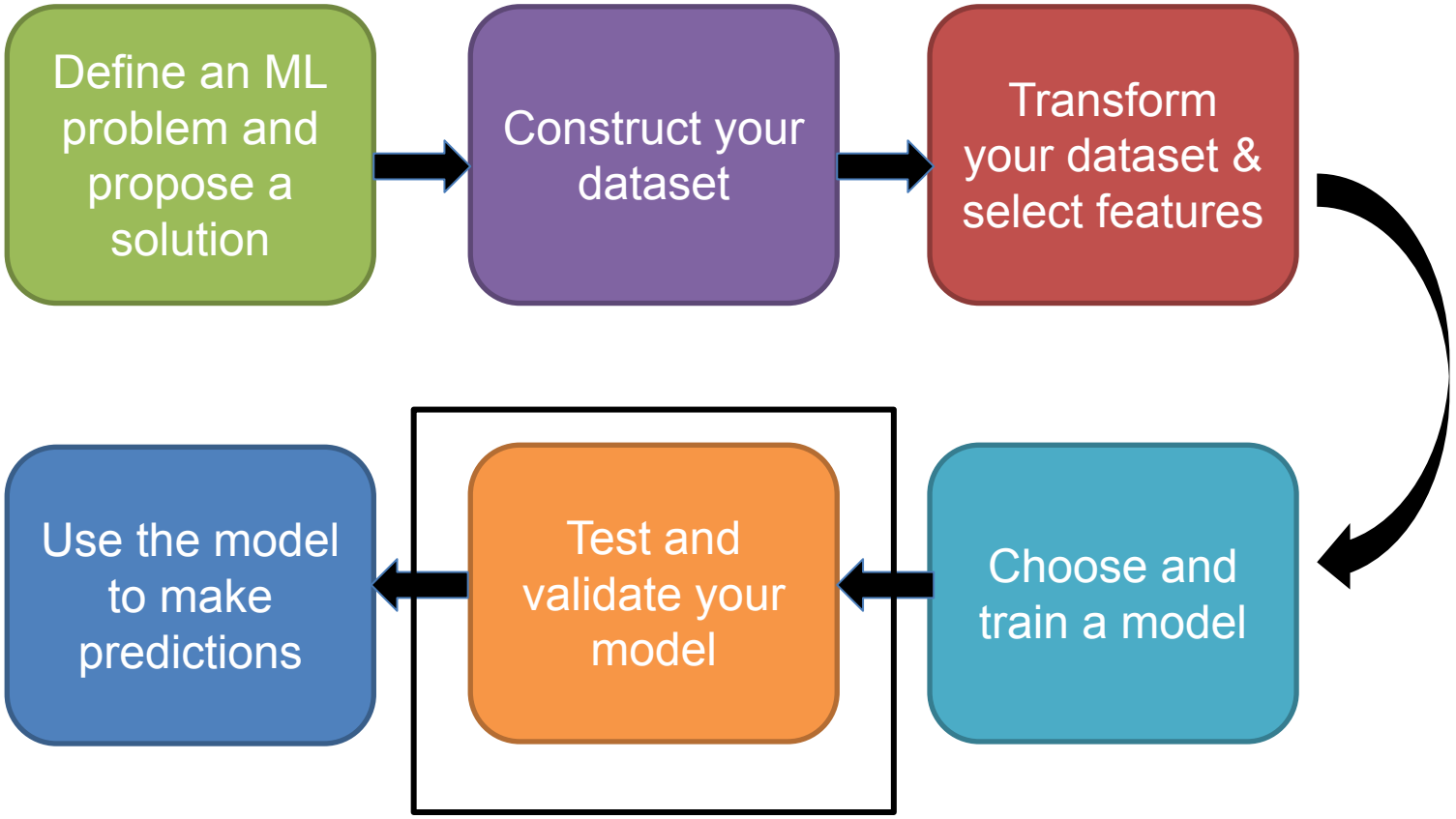


Support Vector Machine

- SVMs are not machines they are algorithms (dumb name)
- Uses a technique called the kernel trick to transform input data and then based on these transformations it finds an optimal boundary or hyperplane between the possible outputs to classify
- Similar to linear discriminant analysis (LDA)
- Can be used for both classification and regression

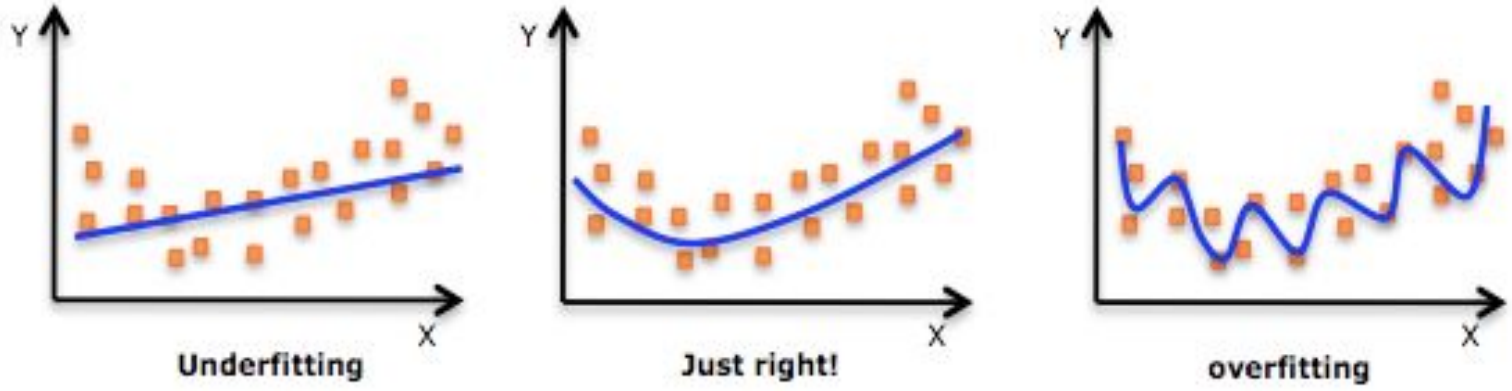


Machine Learning Workflow



Testing & Validating

- Testing and validating ensures your model isn't using too few parameters (underfitting) or too many parameters (overfitting)
- Consequences are:
  - Fitted model is not good for prediction of new data – prediction error is underestimated
  - Model is too elaborate, it models "noise" that will not be the same for new data



Testing & Validating

- Machine learning if subject to problems with overfitting, especially since it is easy to use too many parameters
- EVERYONE falls into this trap
- Key to prevent this is to use:

✔ External validation datasets

✔ K-fold cross-validation

✔ Leave-one out methods

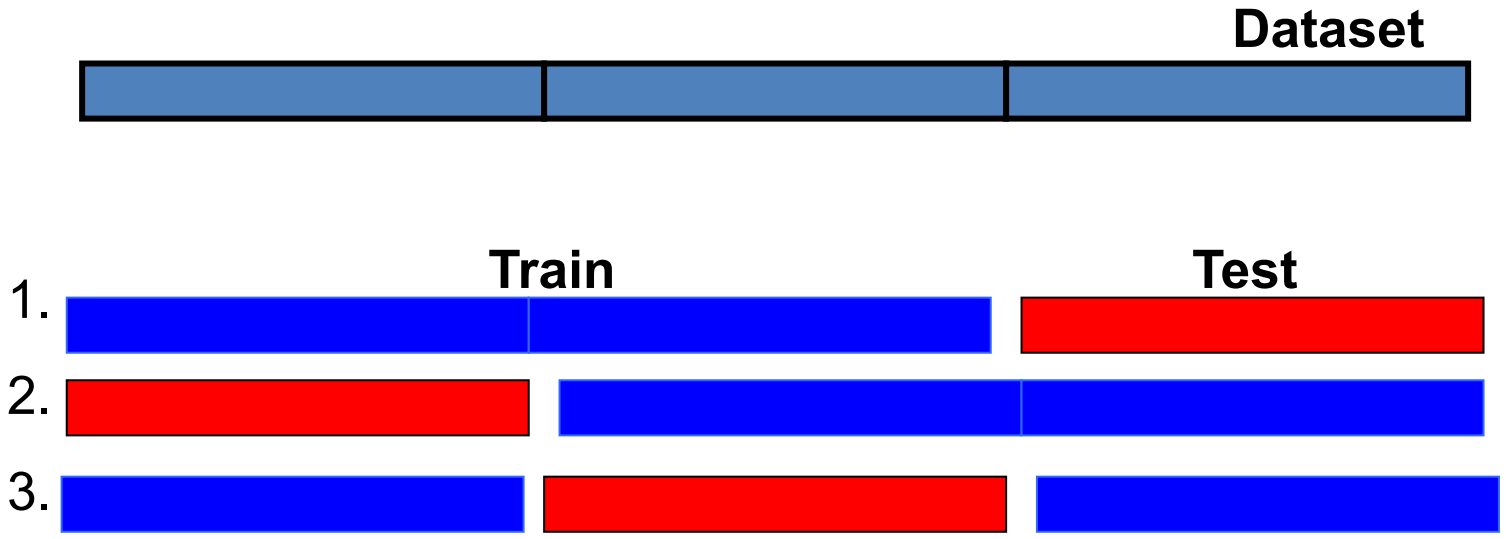
✔ Permutation testing or combinations of all 4 methods

Training/Testing Data

- Often people take their data set and divide it into 2 groups (the holdout method)
- 2/3 (66% of the data) is the training data which is used to train the model
- 1/3 (33% of the data) is the testing data which is “held out” and used to assess the model
- During training the model must not see the test data – EVER
- You can further divide the training data into smaller parts or “folds” to avoid training bias

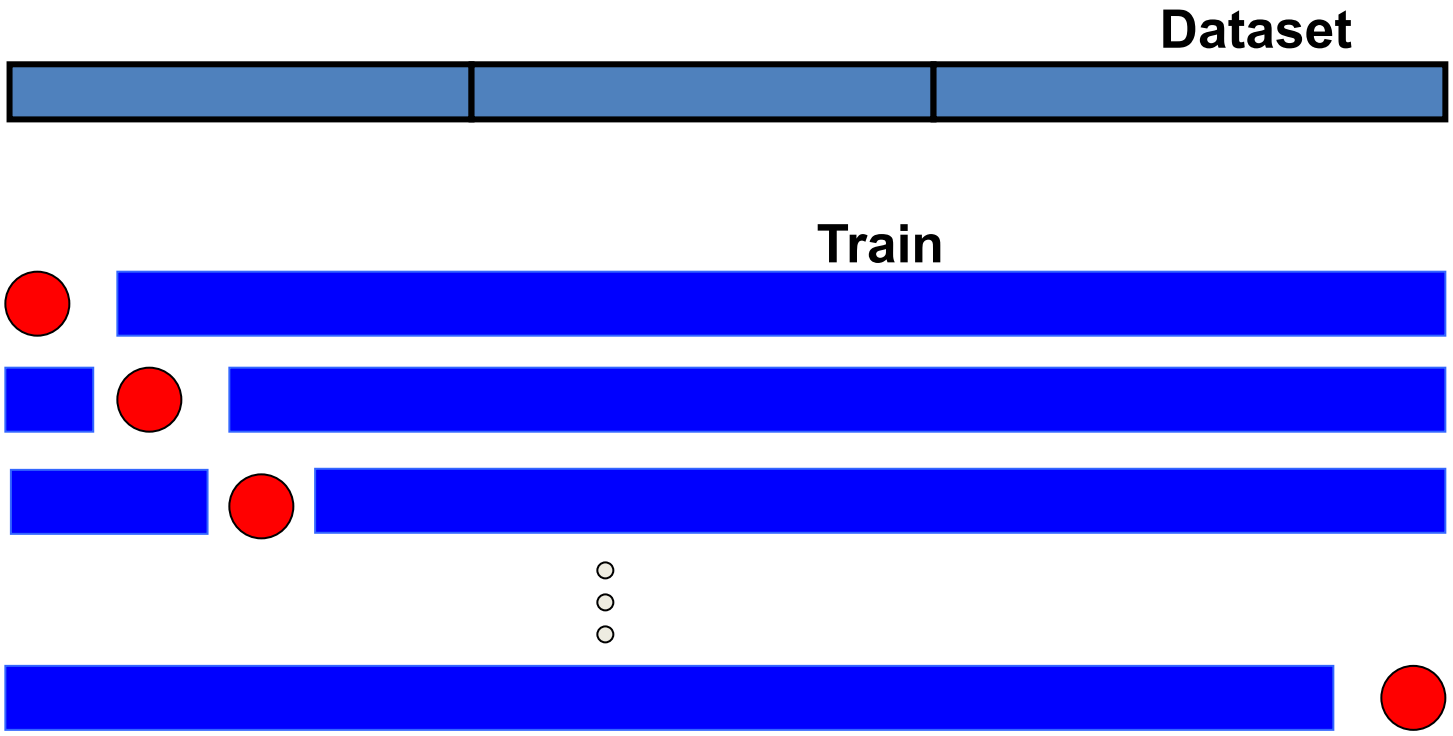
N-Fold Cross Validation

- **Split the dataset into different folds, here is an example of 3-fold cross validation with 3 rounds of testing and training on 1 dataset**

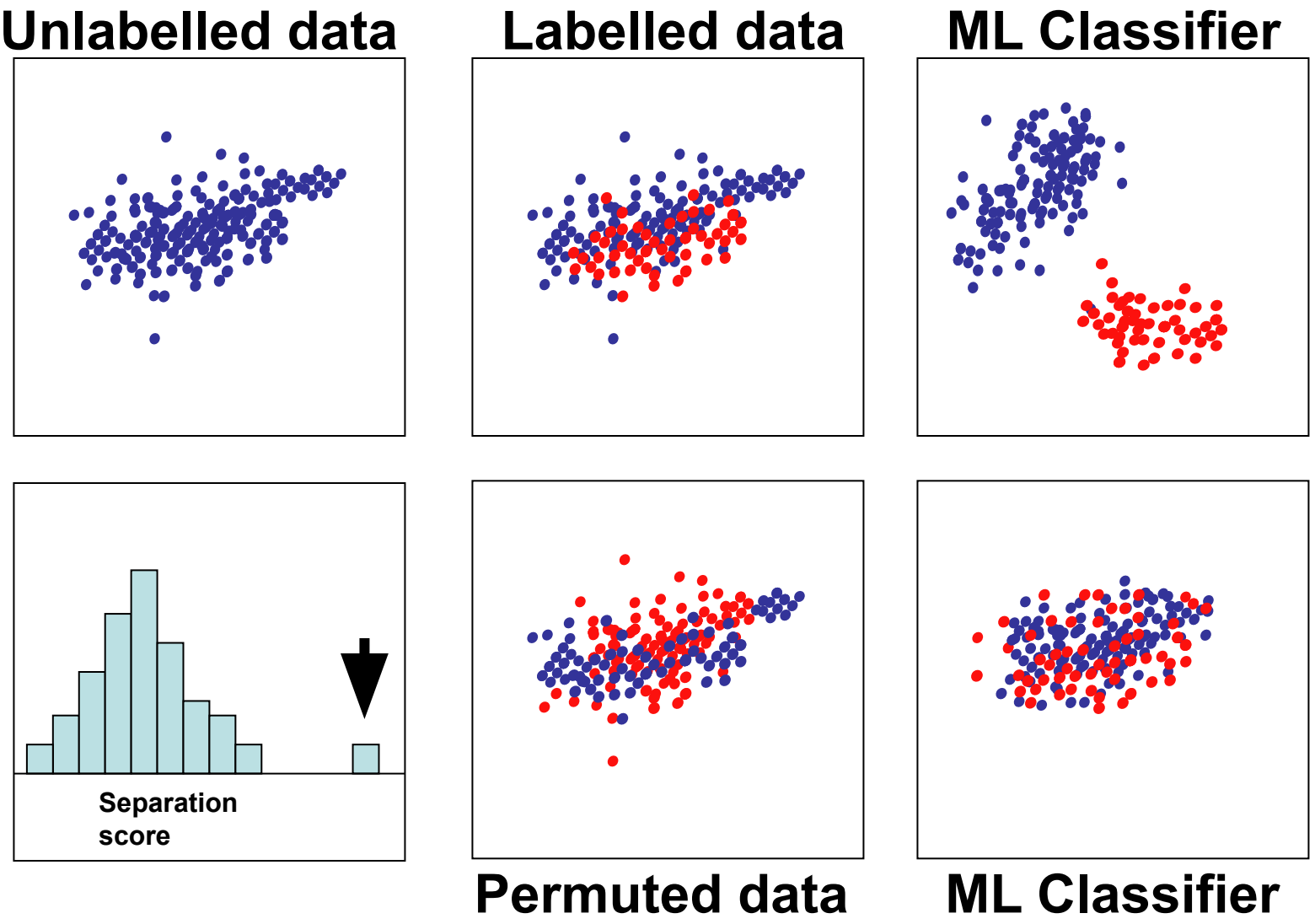


Leave-One-Out Validation

- **Train on all but one example and test on that one, repeat the process until finished**



Permutation Testing



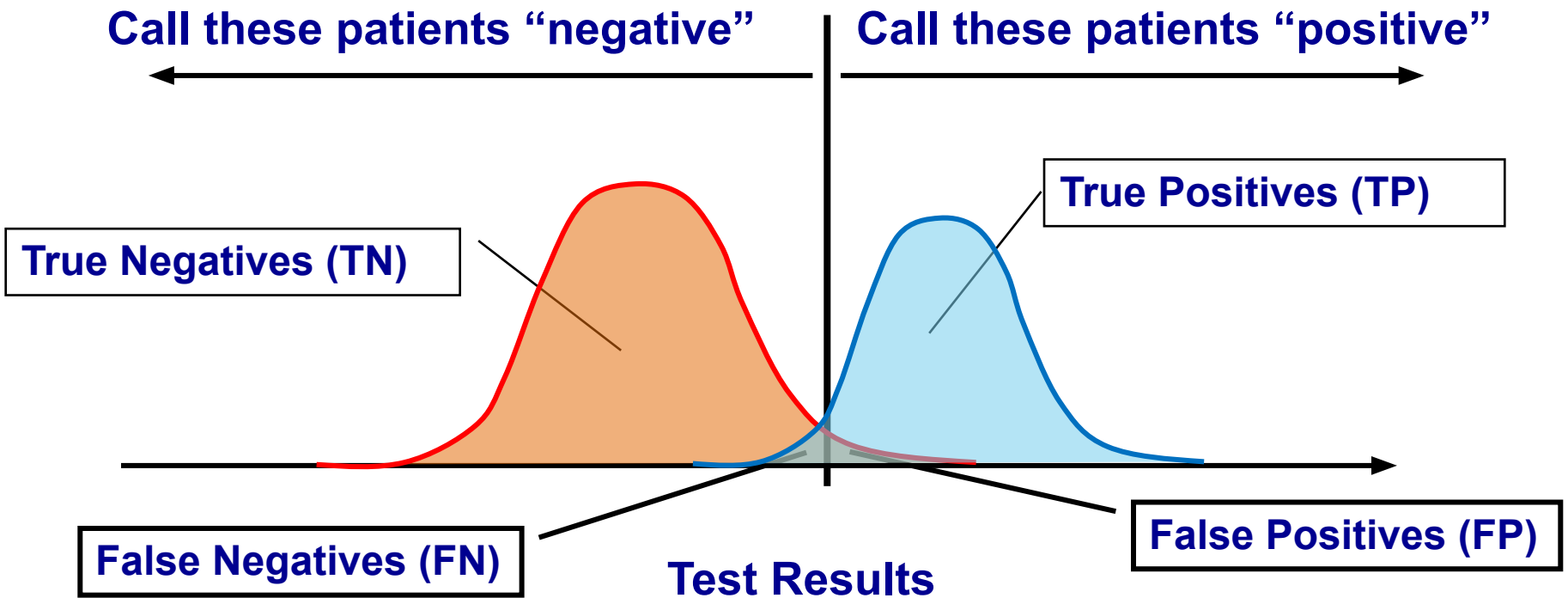
Assessing ML Models – Confusion Matrix

- **Basic concepts**
  - True positives (TP)
  - True negatives (TN)
  - False positives (FP)
  - False negatives (FN)
  - Sensitivity (Sn)
  - Specificity (Sp)

	p' (Predicted)	n' (Predicted)
P (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

- **Sn (sensitivity) = True positive rate**
- **Sp (specificity) = True negative rate**

An Example



Control

Disease

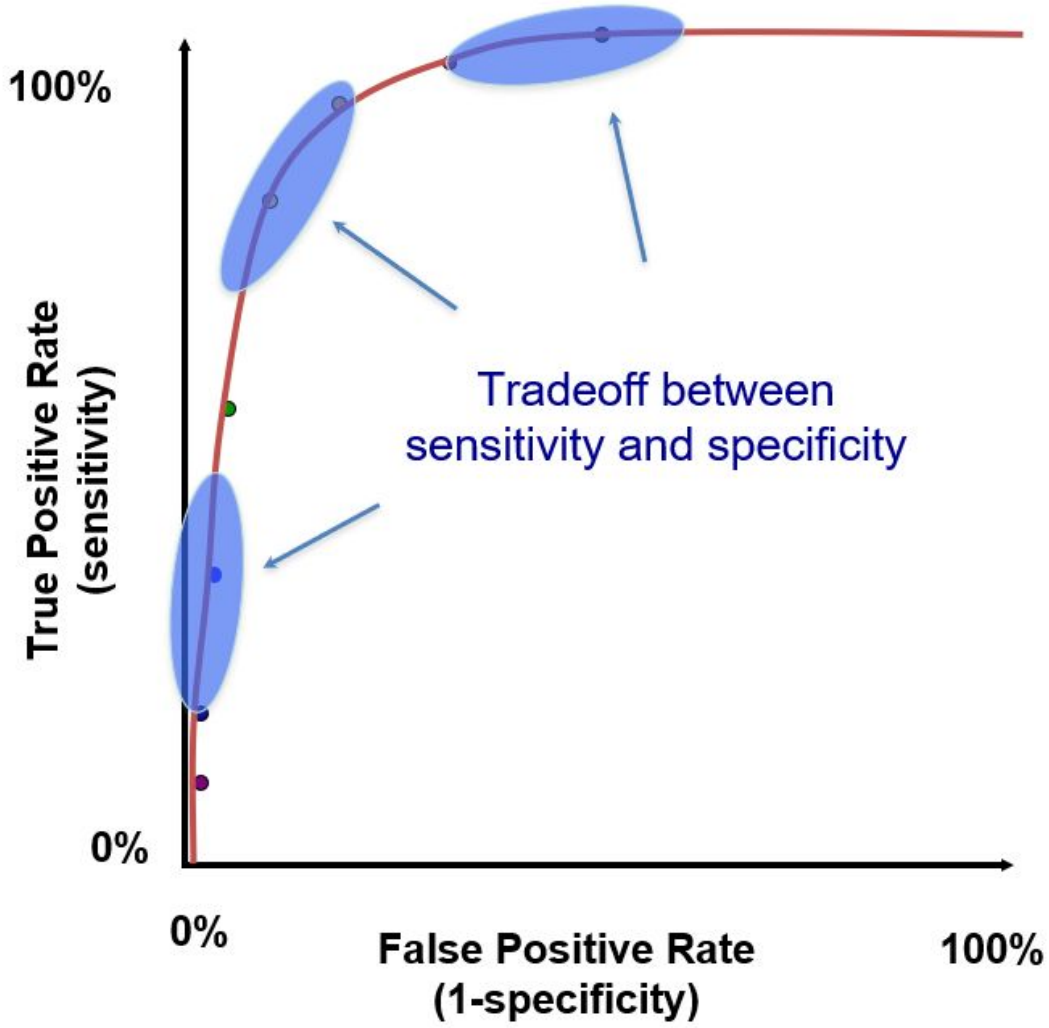
Sn = TP/(TP + FN)

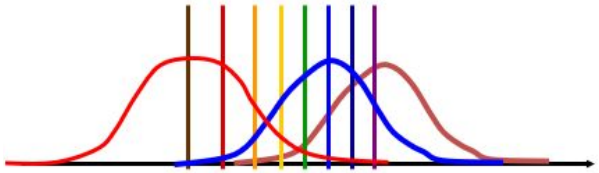
Sp = TN/(TN + FP)

ROC Curves

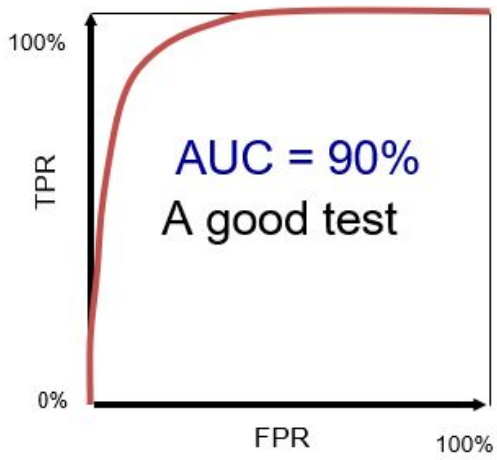
- **ROC = Receiver Operating Characteristic**
  - **A historic name from radar studies**
  - **Very popular in biomedical applications**
    - To assess performance of classifiers
    - To compare different biomarker models
- **A graphical plot of the true positive rate (TPR) vs. false positive rate (FPR), for a binary classifier (i.e. positive/negative) as its cutoff point is varied**

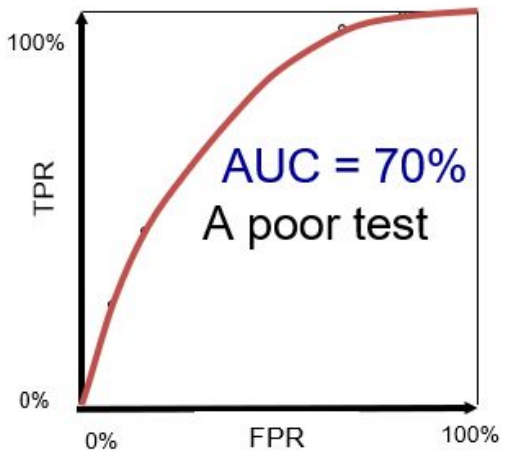
ROC Curve

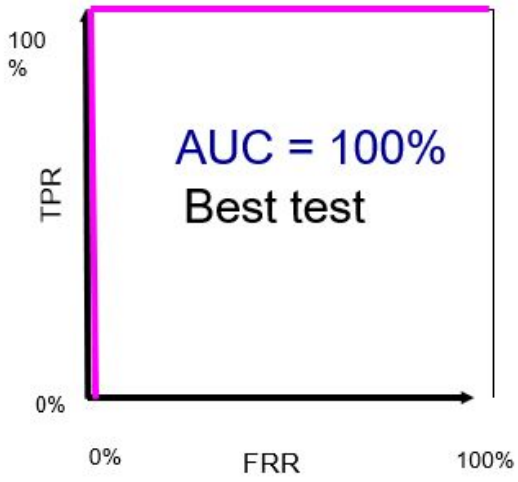

100%
True Positive Rate
(sensitivity)
Tradeoff between
sensitivity and specificity
0%
0%
False Positive Rate
(1-specificity)
100%

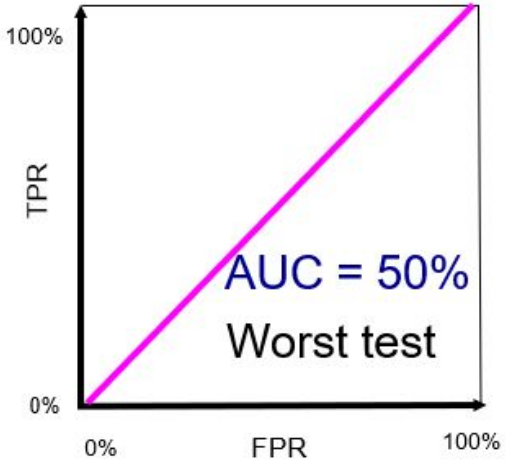


Area Under ROC Curve (AUC)

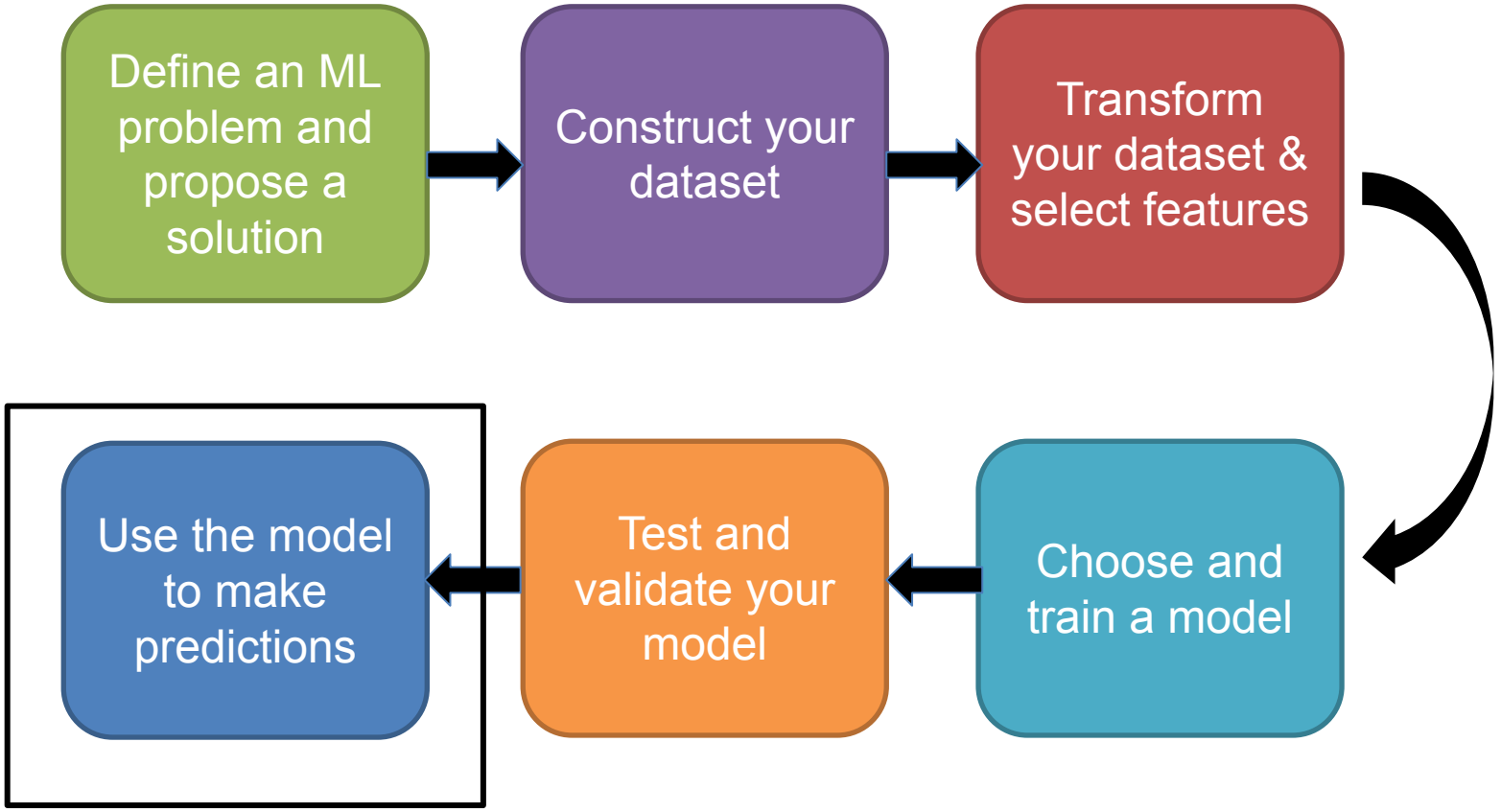




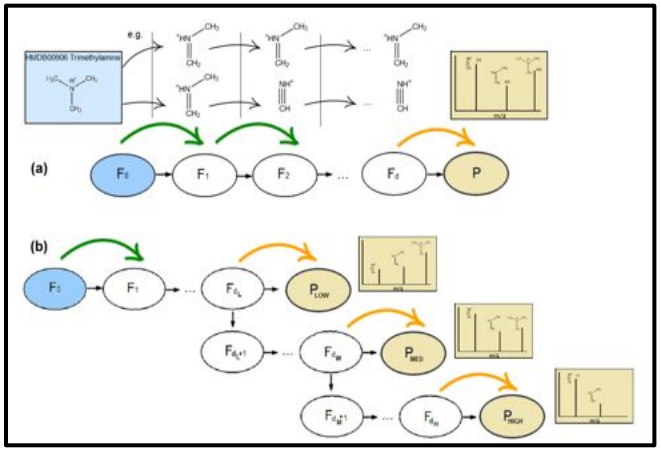


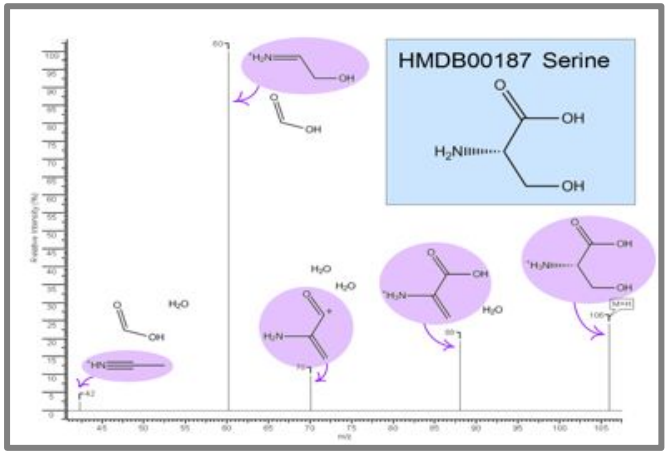


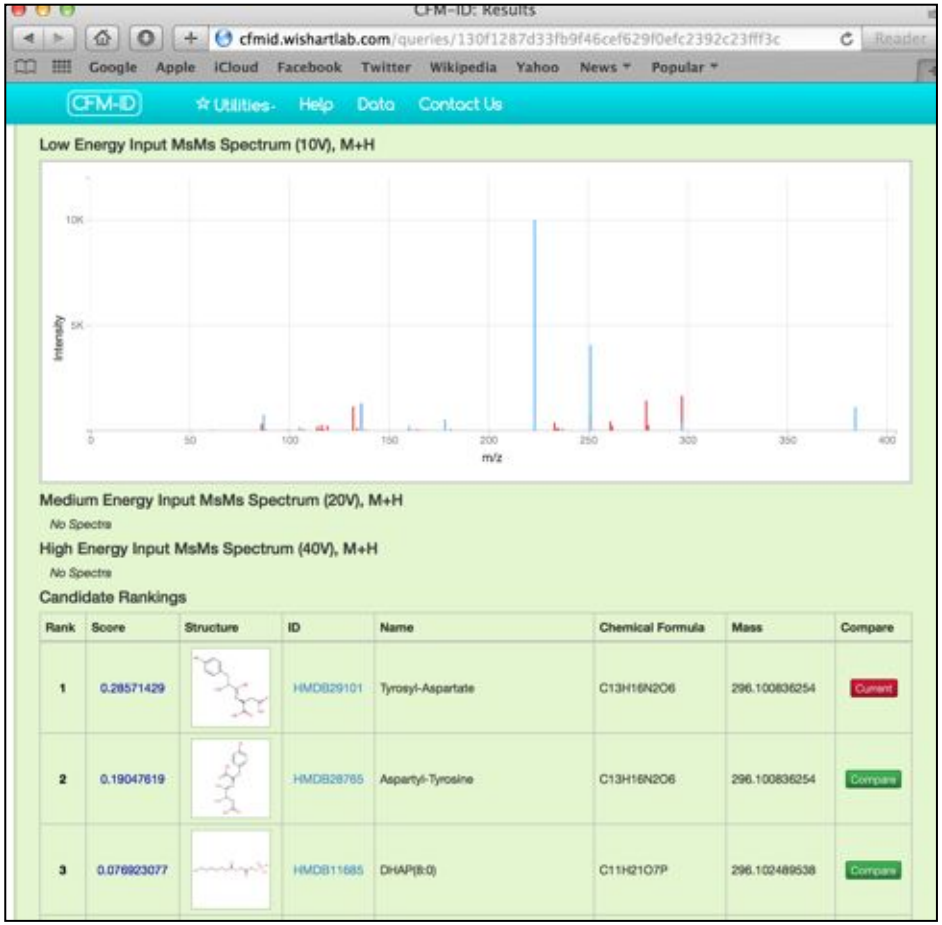
Machine Learning Workflow



Predicting MS/MS Spectra via HMMs & ANNs

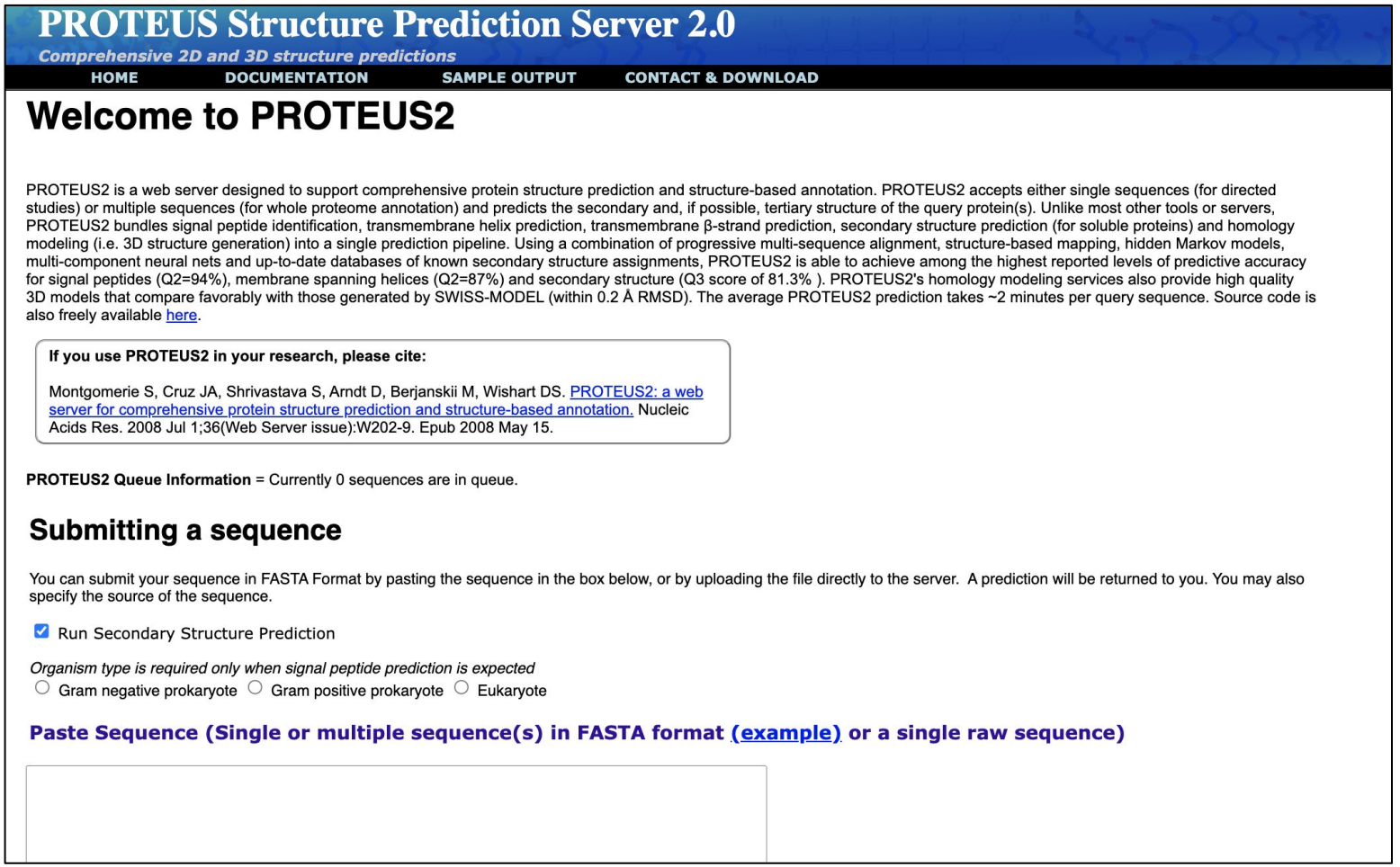






http://cfmid.wishartlab.com

Predicting Secondary Structure with ANNs



http://www.proteus2.ca/proteus2/

Outline/Course Rationale

- Introduce students to 2 types of machine learning models (DT and ANN)
- Apply them to 3 bioinformatics or biological problems (general classification, secondary structure prediction, gene finding)
- Deep dive into the algorithms and code using Python and Google Colab (need to understand how a car works before you can drive)
- Show how same ML tasks/problems can be coded quickly via Keras and SciKit-Learn

End of Module 1
(jump to slide 111)

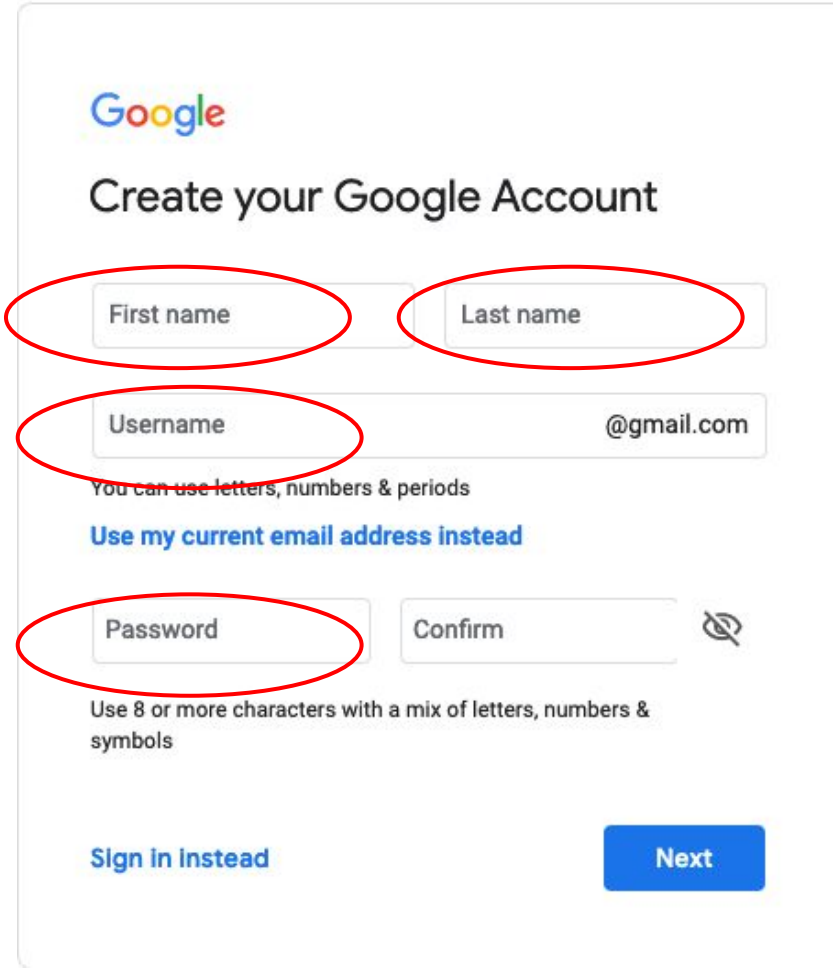
Google Colab

- Colab is a Python development environment that runs in the browser using Google Cloud
- Need to have a google account/google drive (see next slide)
- No need to install or run Python locally or deal with local OS issues
- Allows you to access, import and edit previously written Python code
- Allows you to share, save and download code
- Allows easy access to NumPy
- Allows easy access to GPUs, ML code and TensorFlow for ML applications

How to Set up a Google Account

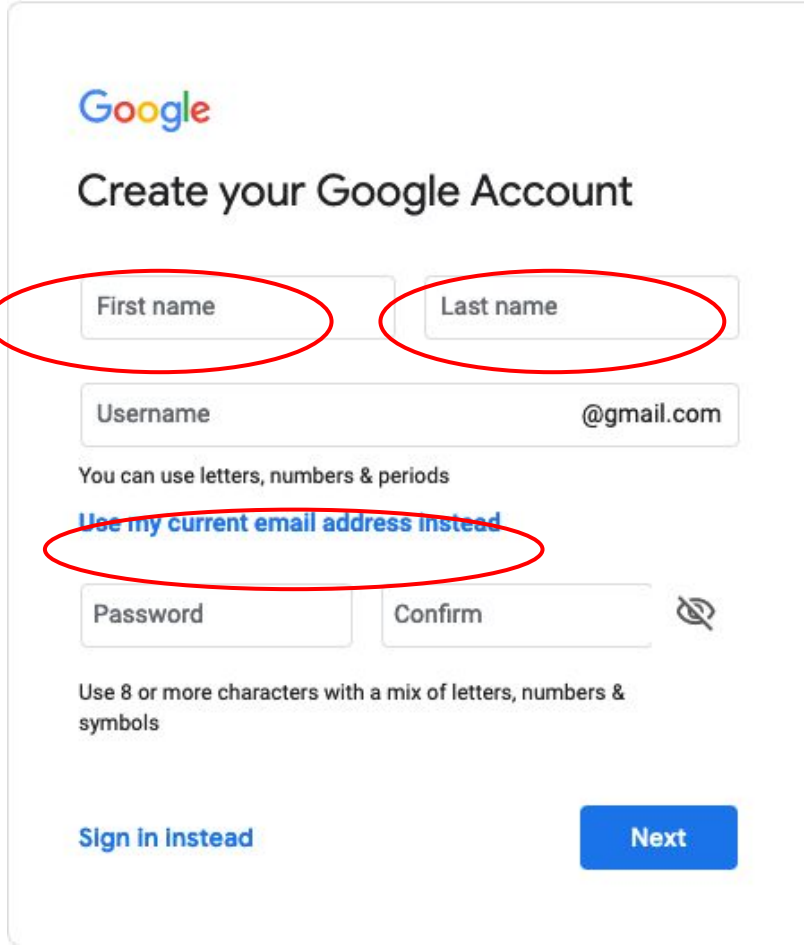
How to Set Up a New Google Account and New Email (gmail)

- To create a Google account go to Google account creation page
- Enter First name (circled red) and Last Name (circled red)
- Enter new "Username" you want for you account (circled red)
- If your username taken, the box will turn red, and you must choose Different username
- Type in a password and confirm it
- Click Next
- Verify your mobile phone number with the code sent through text message
- Click Verify



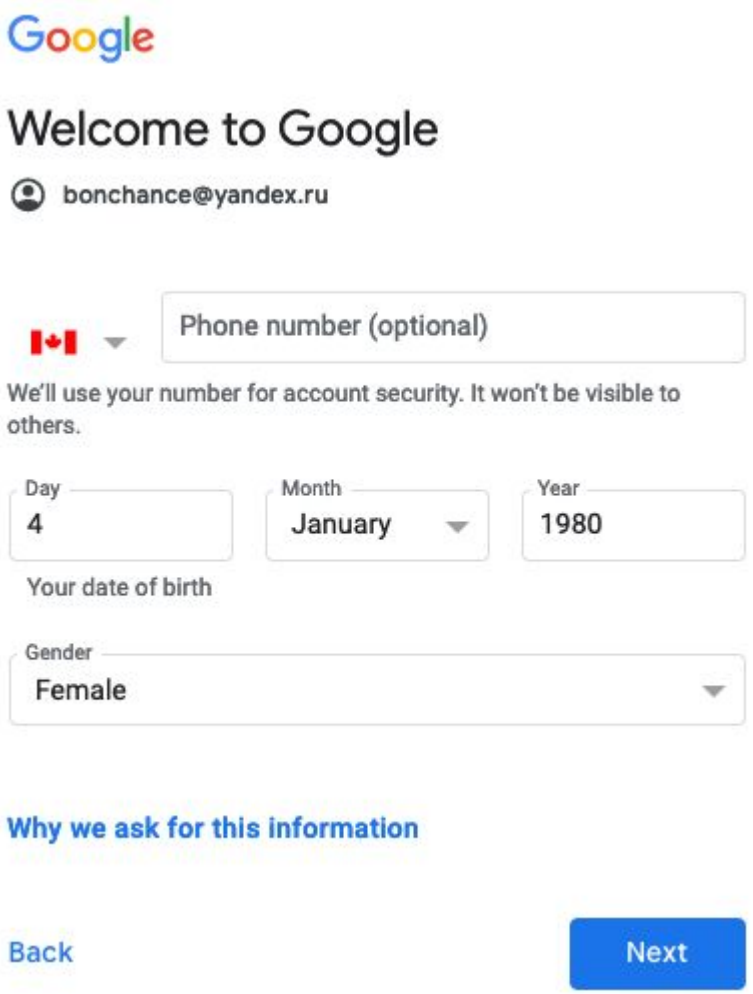
How to Set Up a New Google Account Using Your Existing Email (Not Gmail)

- To create a Google account go to Google account creation page
- Enter First name (circled red) and Last Name (circled red)
- Enter "Use my current email address instead" (circled red)
- Enter your current email address
- Click Next
- An email will be sent to your email address
- Go to your email account
- Open an email from Google "Verify your email account"
- You should see the code sent to your email
- Enter the code sent to you and click 'Verify'



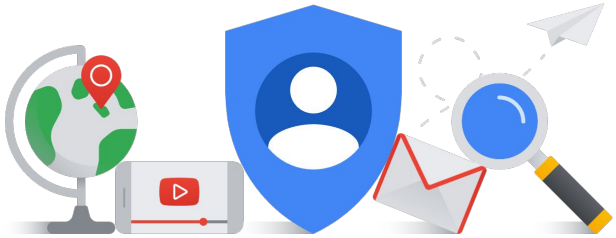
How to Set Up a New Google Account Using Your Existing Email (Not Gmail)

- Open your email
- You should see an email from Google “Verify your Google account”
- Verify your email address with the code sent to your email
- Enter the code sent to you
- Click Verify
- Enter your phone number (optional)
- Enter your day/month/year of your birthday
- Select your Gender
- Click Next



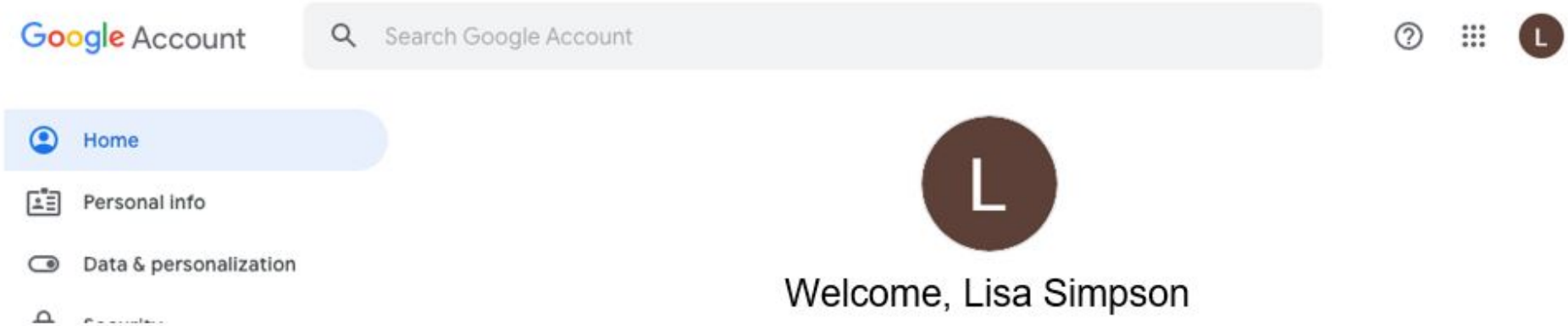
How to Set Up a New Google Account Using Your Existing Email (Not Gmail)

- Google will present the terms, conditions, and privacy policies for your Google Account
- Once you've read everything over, click I agree



How to Set Up a New Google Account Using Your Existing Email (Not Gmail)

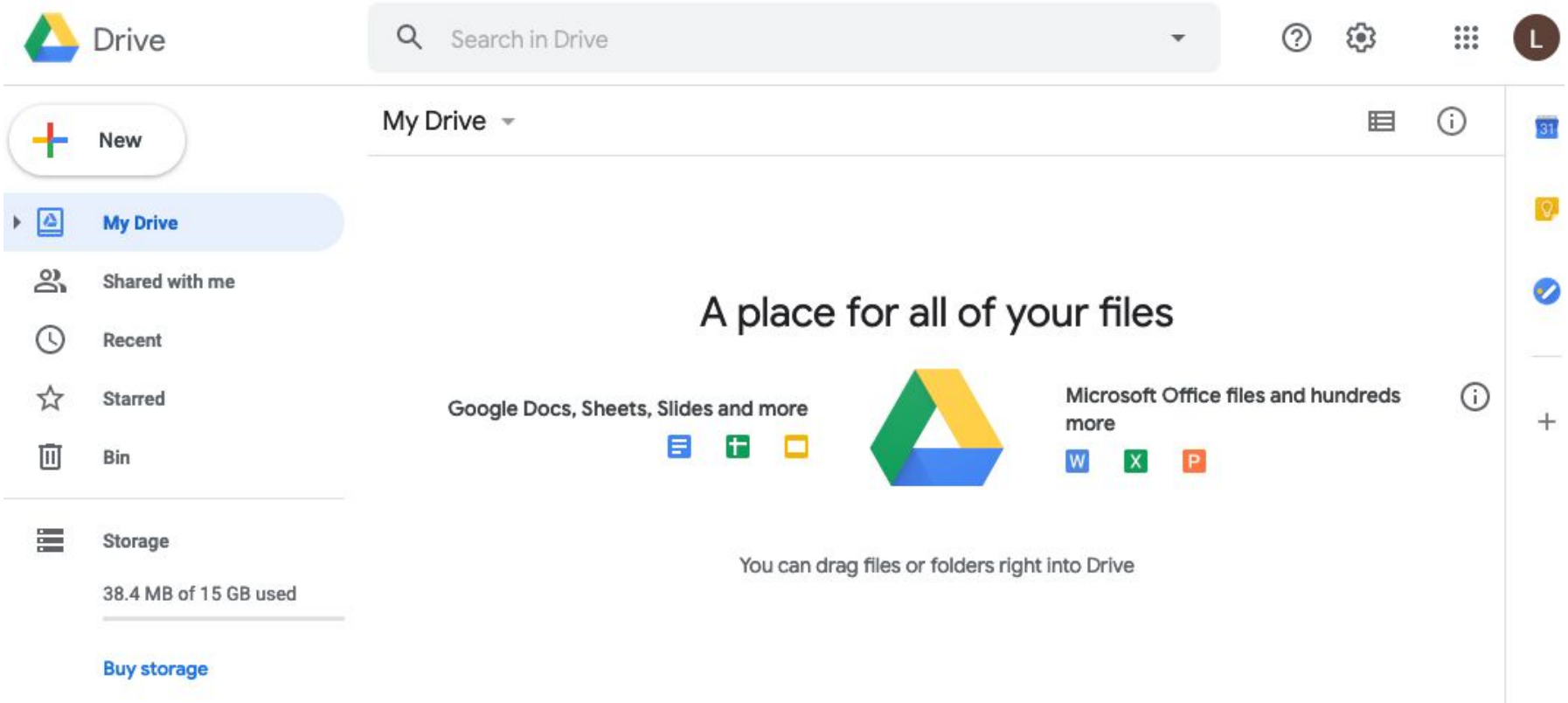
Your new Google account will look like this picture below with your name



Google Colaboratory

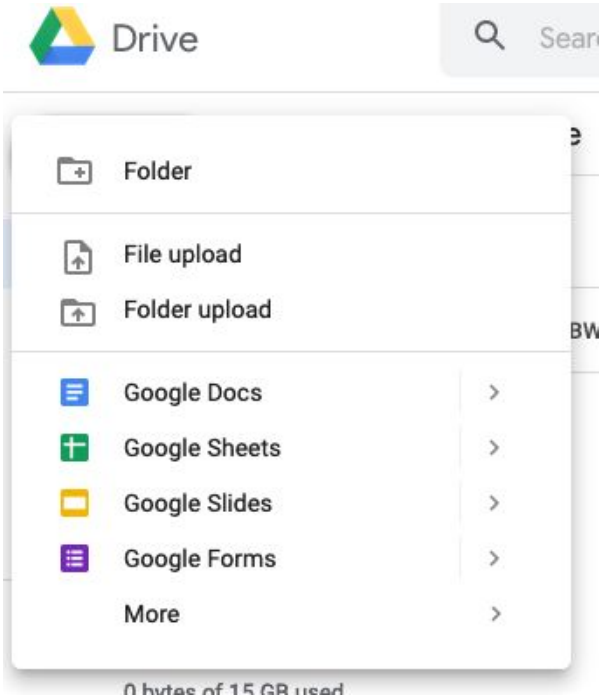
- Access by typing “Colab” in Google or go to https://colab.research.google.com/
- Click on “Welcome to Colaboratory” to launch
- Click on the “Introduction to Colab” video to learn more (3 minutes)
- Colab home page has File/Edit/View/Insert functions on upper left of page
- Runs like a Jupyter notebook
- Colab notebooks allow you to combine executable code, rich text, images, HTML, etc.

How to Work with Colab Files



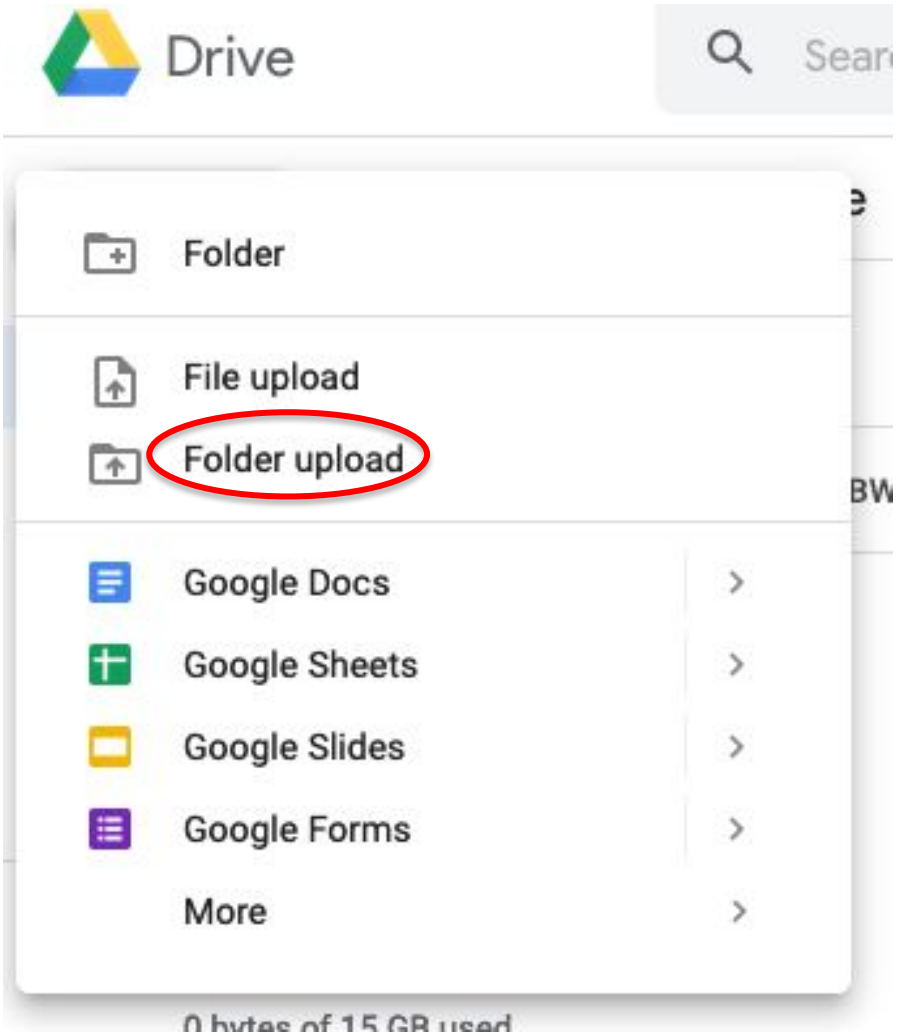
Download CBW Files

- Download the “CBW-MachineLearning-2023” folder into your computer
- Extract the zip file
- Upload the extracted files into your Google drive
  - Go to your main Google drive page
  - Go to the left top Click 



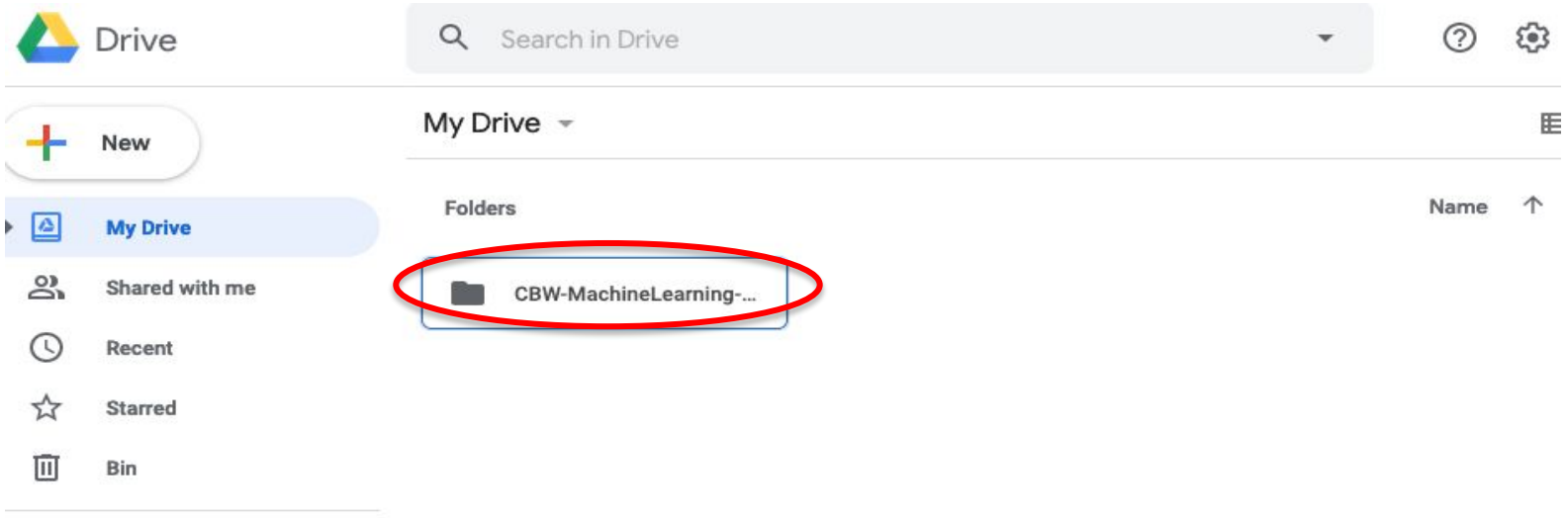
Upload CBW Files

- **Choose Folder upload**



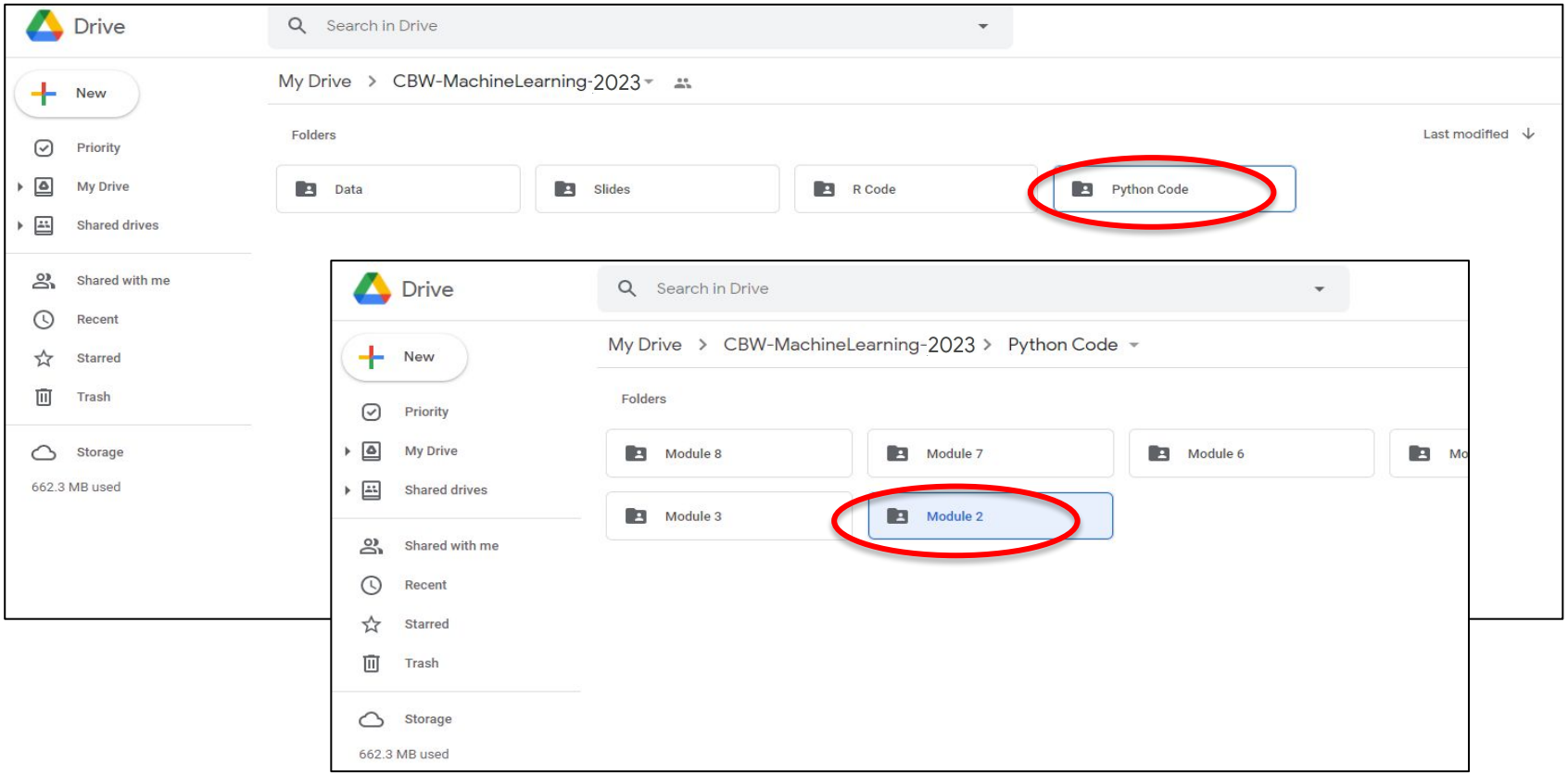
Upload CBW Files *cont...*

- The window will open asking to choose the folder
- Find “CBW-MachineLearning-2023” folder
- Click “open”
- You should see the folder in your drive



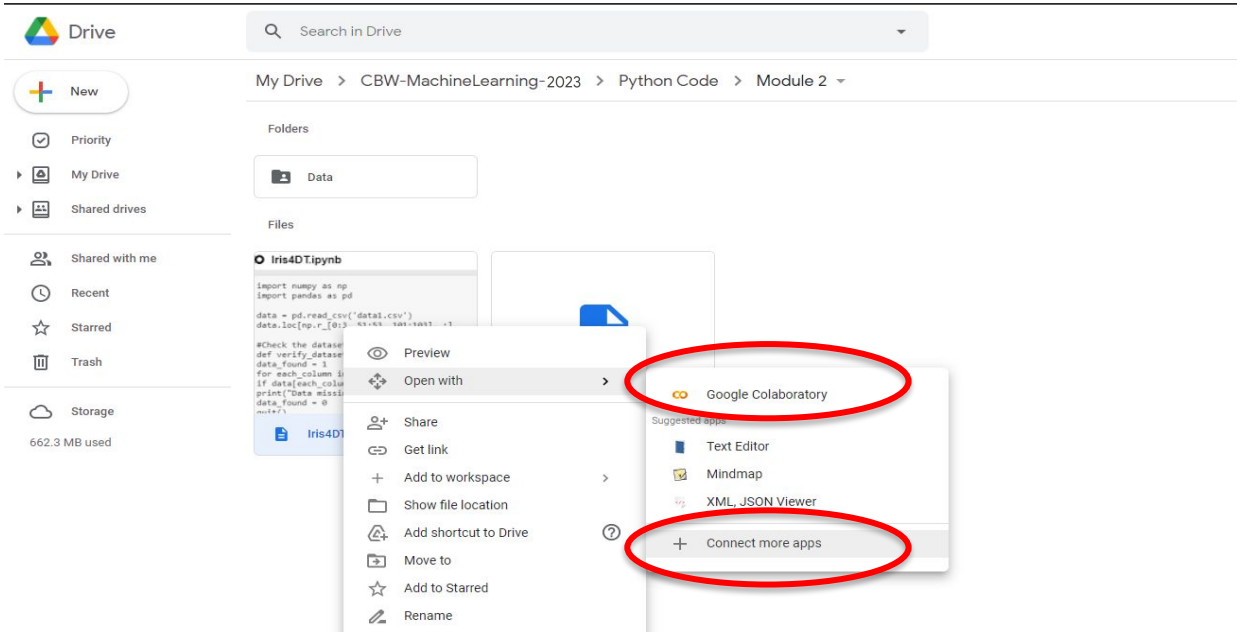
Open the Colab File

- Double Click on the ‘Python Code’ folder
- And the ‘Module 2’ folder



Open the Colab File *cont...*

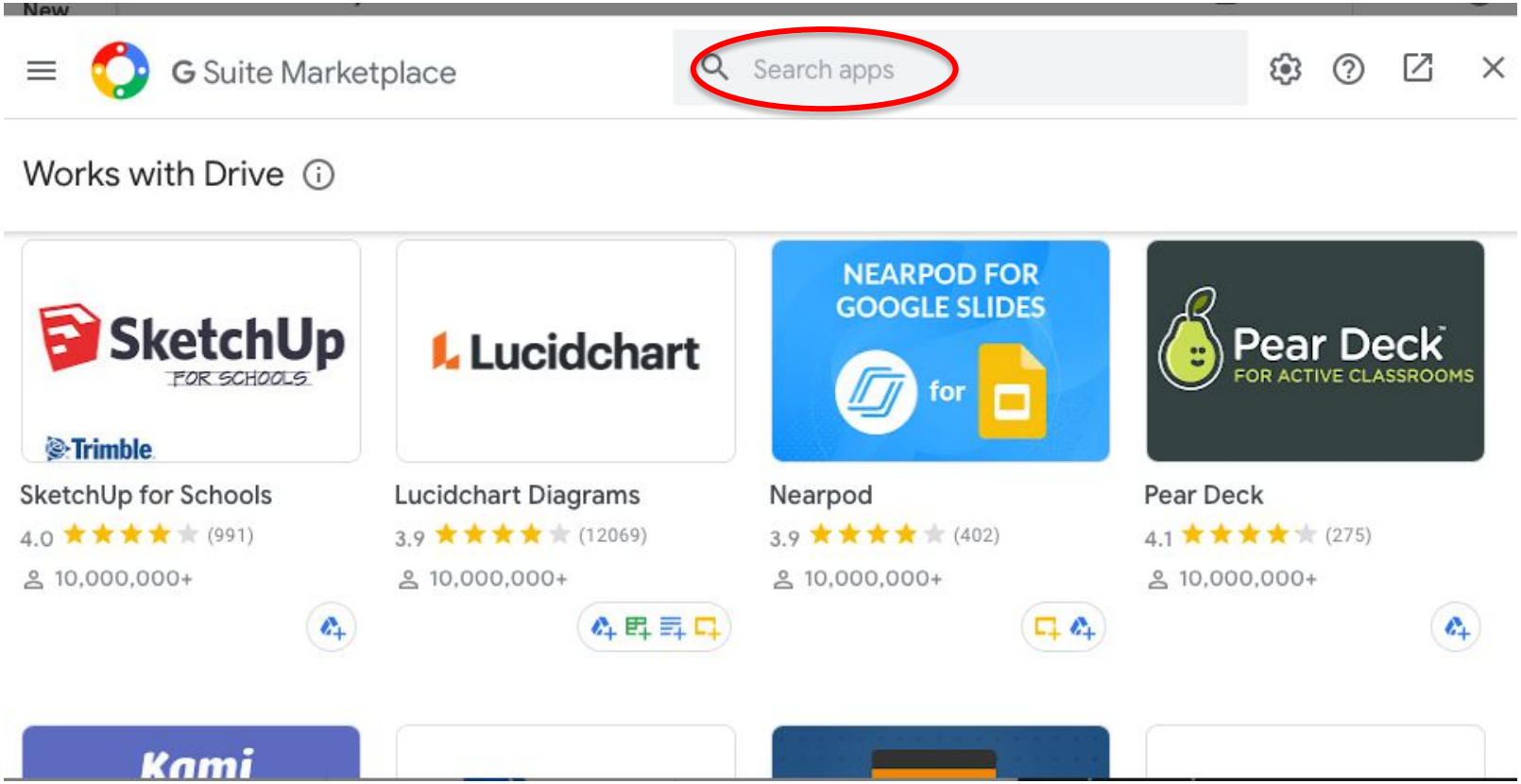
- **Right Click on a file you want to open (for example, 'Iris4DT.ipynb')**
- **Click "open with"**



- **If you have it, Click 'Google Colaboratory'**
- **Otherwise, Click '+ Connect more apps'**

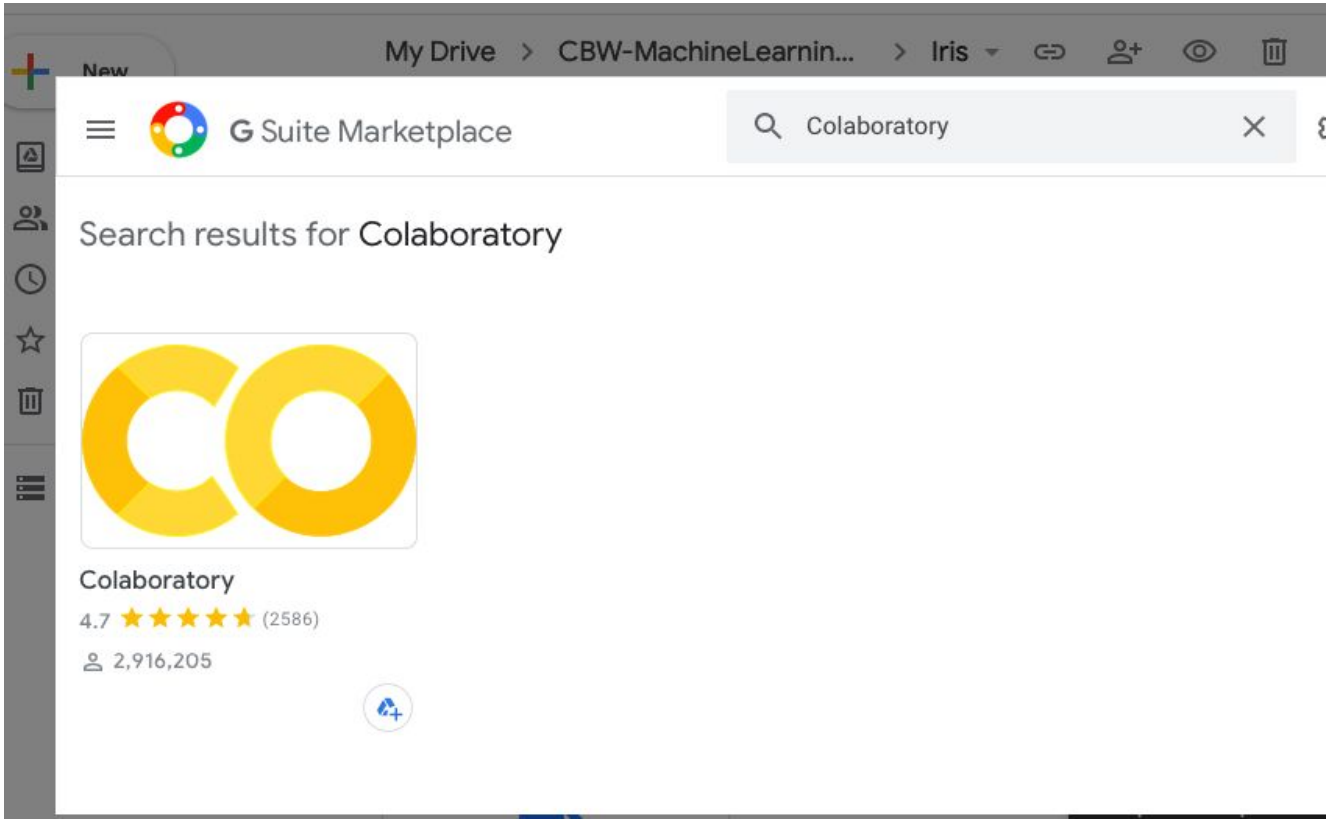
Install the Colab App

- If you have clicked '+ Connect more apps'
- In "search apps" type Colaboratory



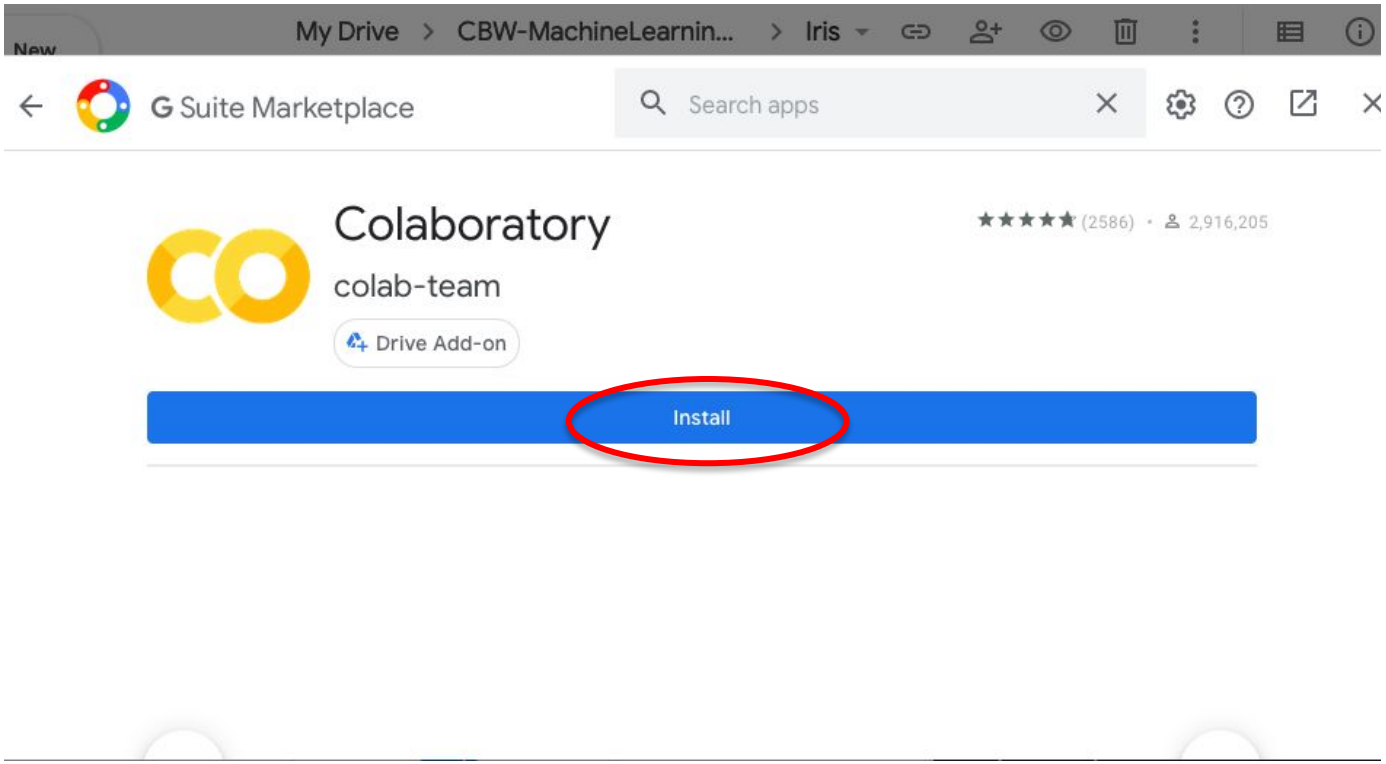
Install the Colab App *cont...*

- **Click on Colaboratory (Colab) to add it to your G Suite**



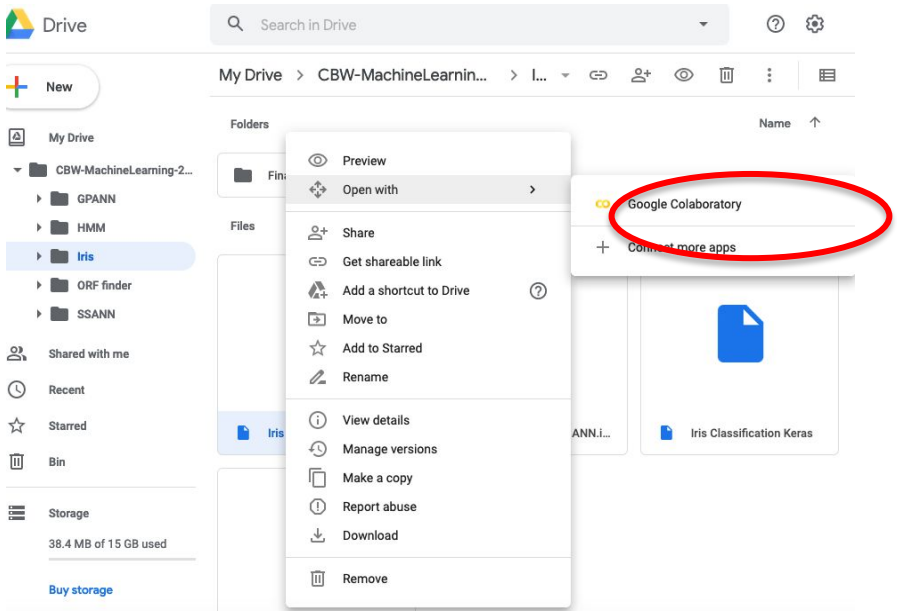
Install the Colab App *cont...*

- **Click install**

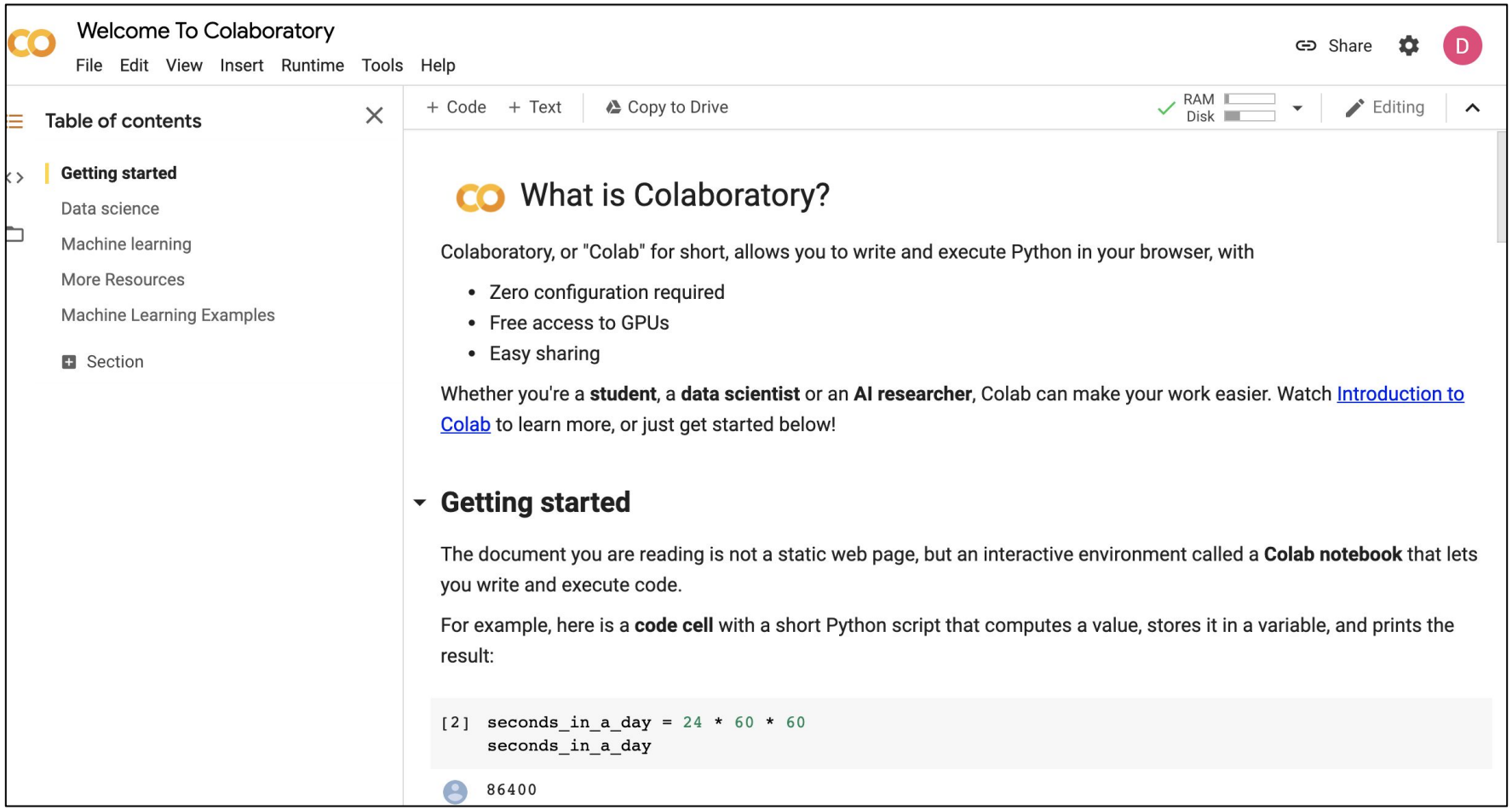


Open a Colab File

- Now right click on any file you want to open with Colab
- Open with Google Colaboratory
- The Colab file will be opened in the Google Colab



Colab Main Page

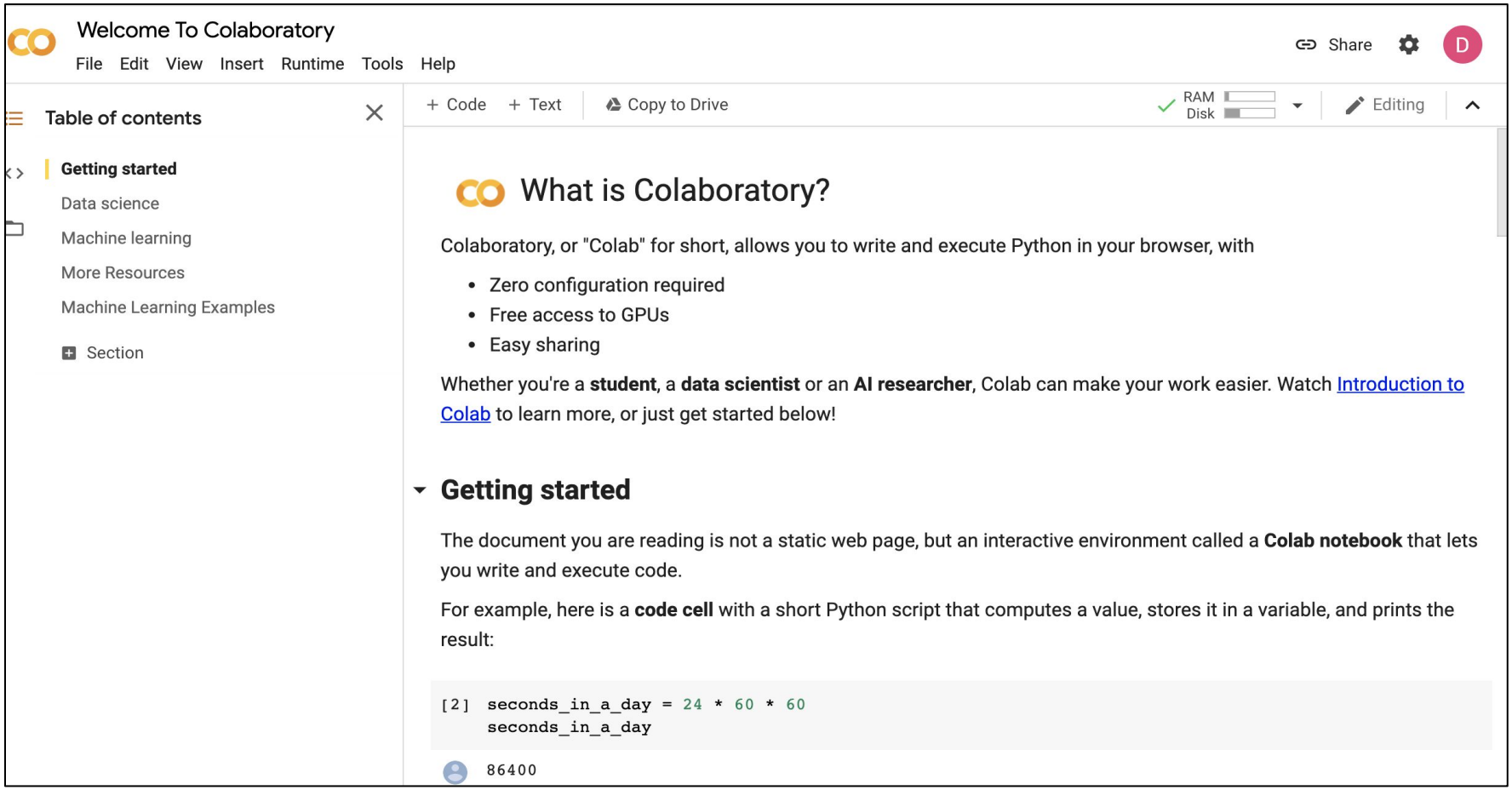


Your First Google Colab Notebook

- Note: Colab uses Google Drive for storing your notebooks, ensure that you are logged in to your Google Drive account before proceeding further
- Step 1: Open the following URL in your browser (you must be signed into your Google Drive): https://colab.research.google.com/notebooks/welcome.ipynb

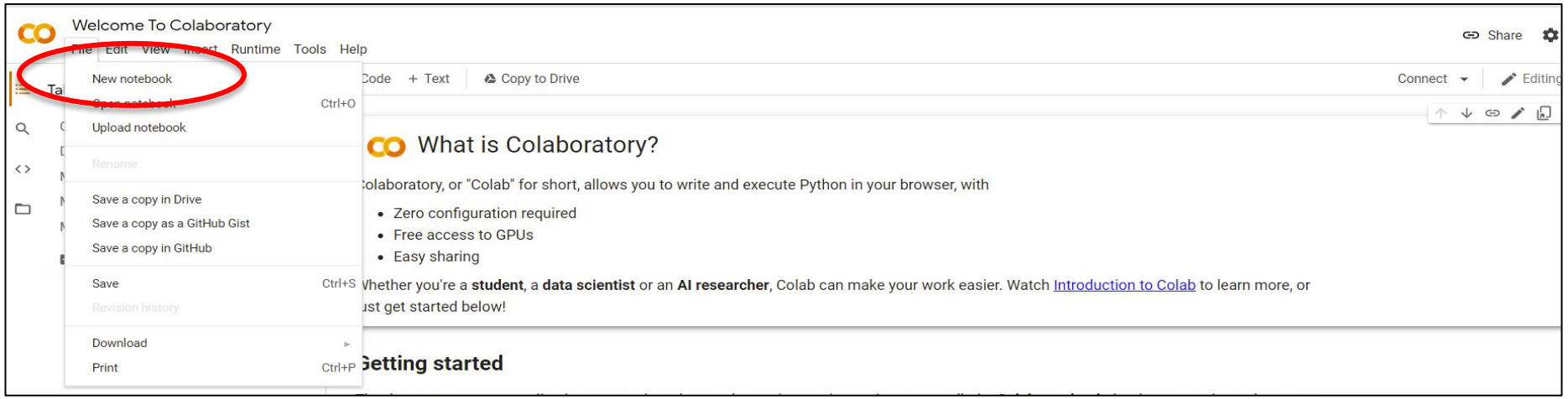
Your First Google Colab Notebook

- **Step 1: your browser will display the following screen**



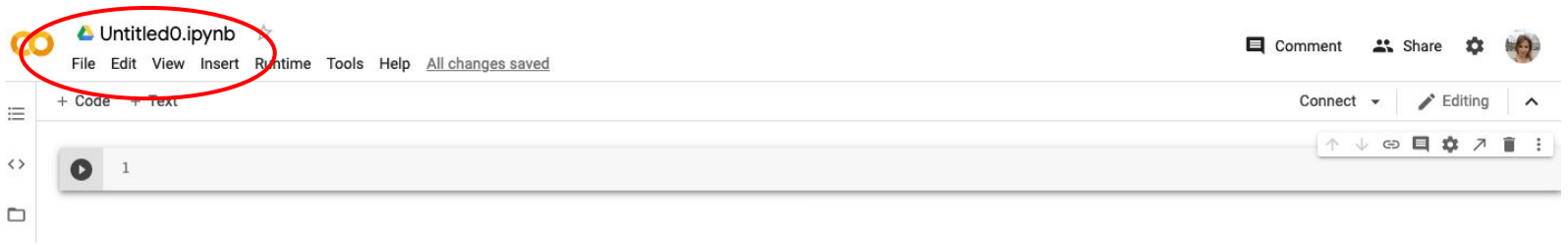
How to Open a New Notebook

- **Step 2: Under File, click New Notebook**
- **A new notebook should open up as shown below**



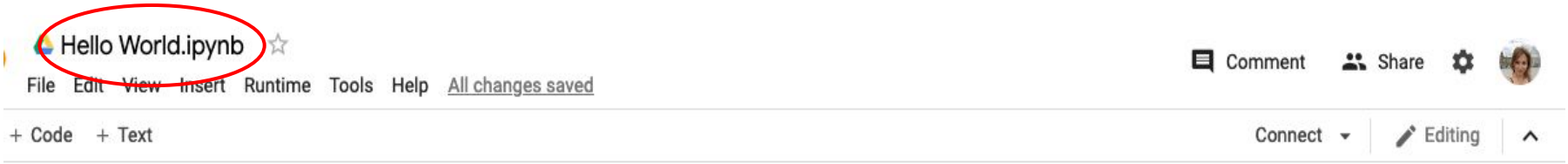
Set a Notebook Name

- By default, the notebook uses the naming convention Untitled0.ipynb
- To rename the notebook, click on this name and type in the desired name in the red circle as shown below



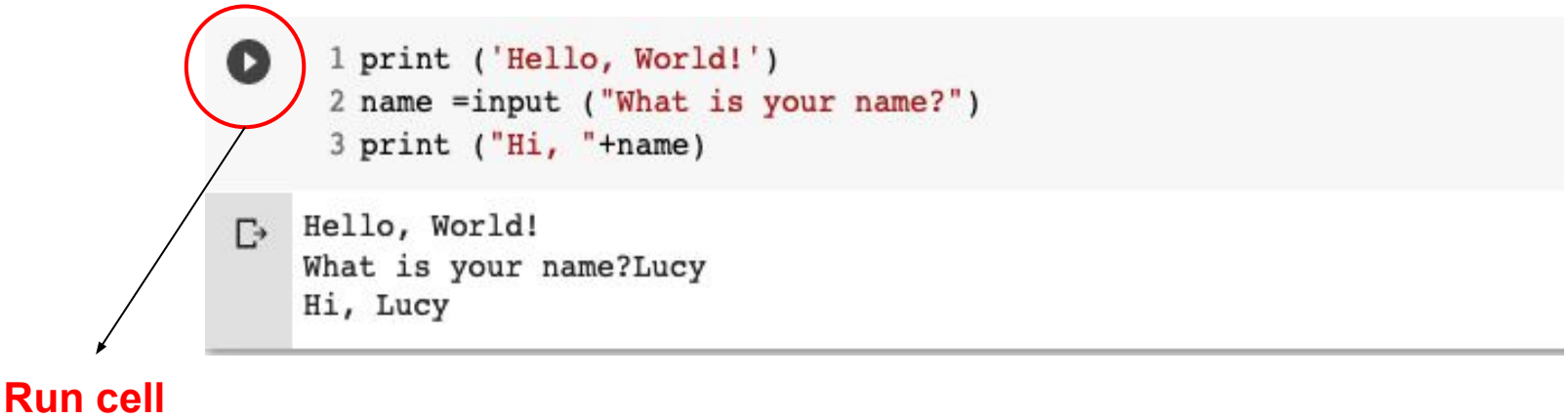
Hello World

- Let's call it "Hello World"
- Type that name in the circle and press Enter
- The notebook will acquire the name you have given now



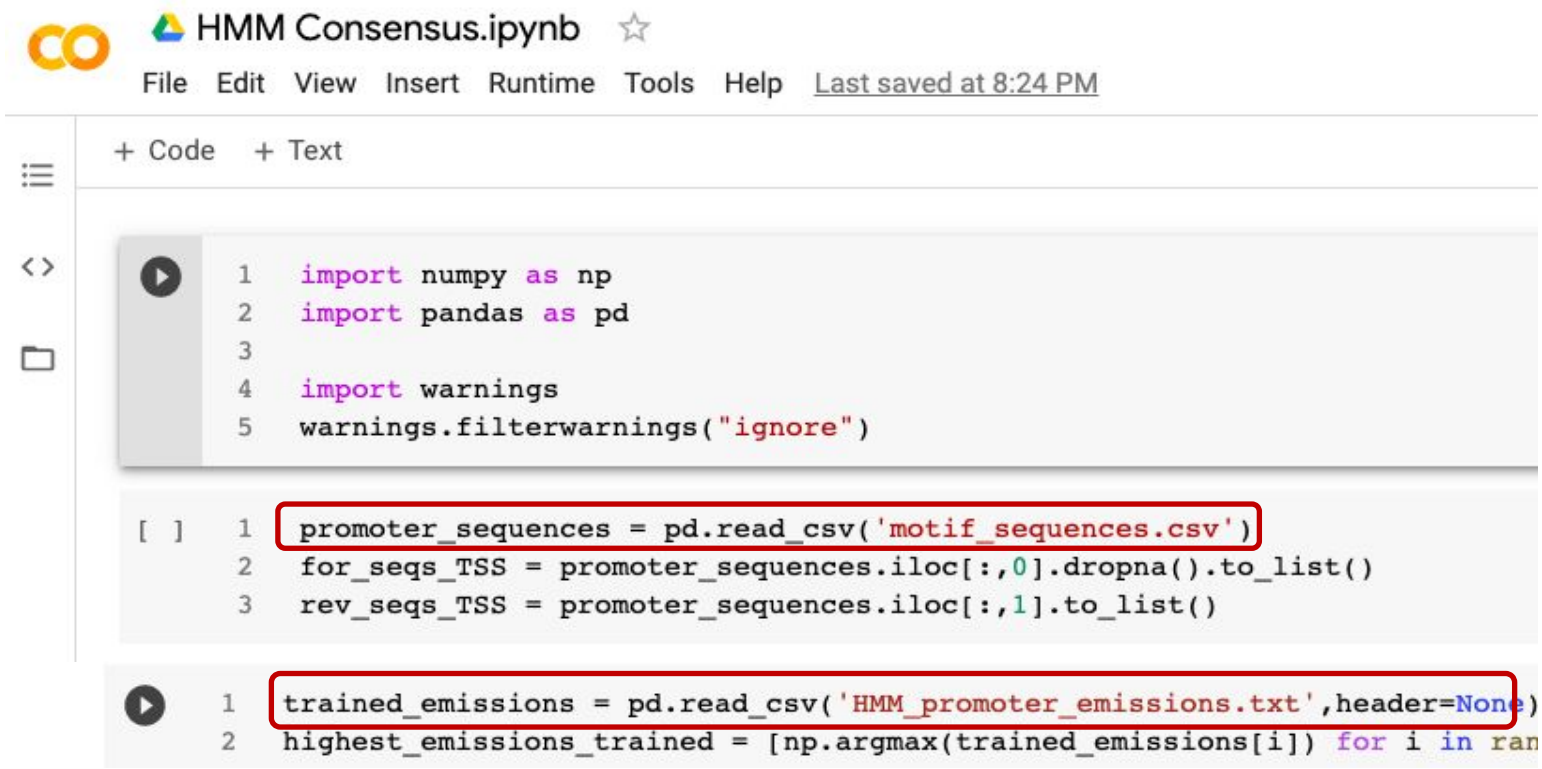
Writing and Executing the Code

- Enter the trivial Python code and run it
- We will show how the simple code “Hello World” can be written
- Write this few lines of code in code cell
- Press run cell
- The output are the results



Uploading the Data Files into the Colab Notebook

- **The files we need for the example code below will be read by pandas in two instances:**

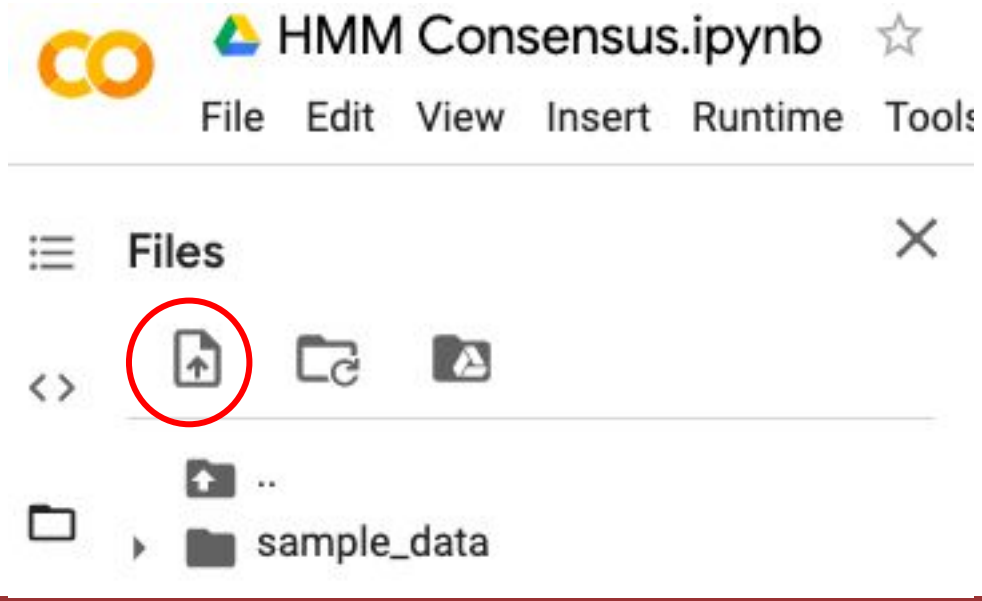


Uploading the Data Files *cont...*

- **Click on the file folder**

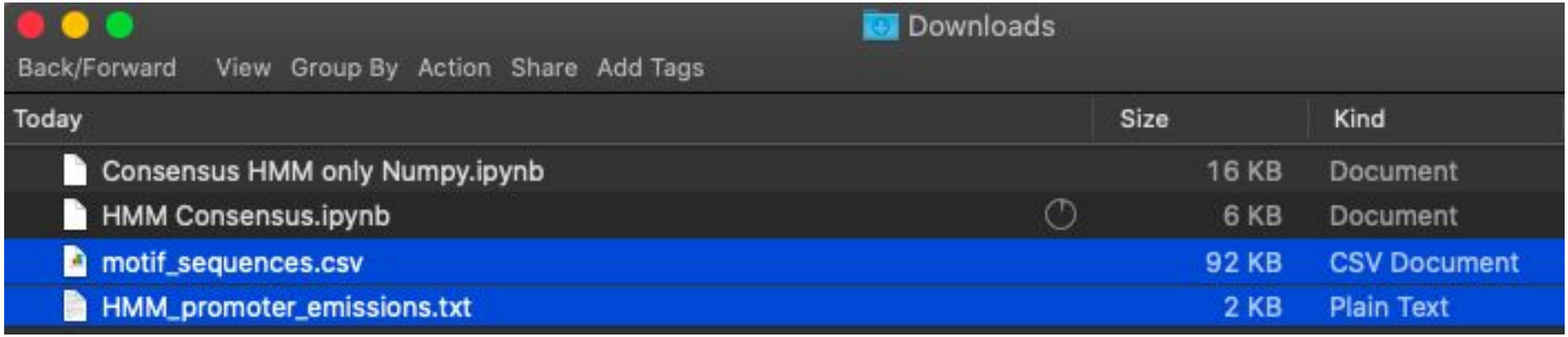
Uploading the Data Files *cont...*

- The code window will shift right and display the options below
- We will upload files (though you can use google drive too)
- Click to select data files



Uploading the Data Files *cont...*

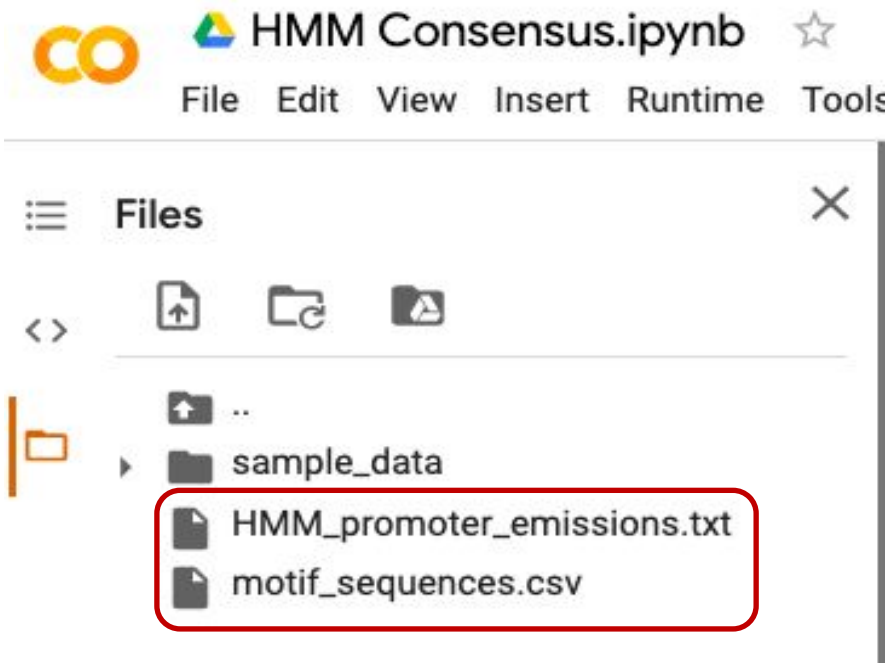
- Browse to where your data files are located
- You can shift+click to select both at the same time
- You can also drag and drop



- Click open

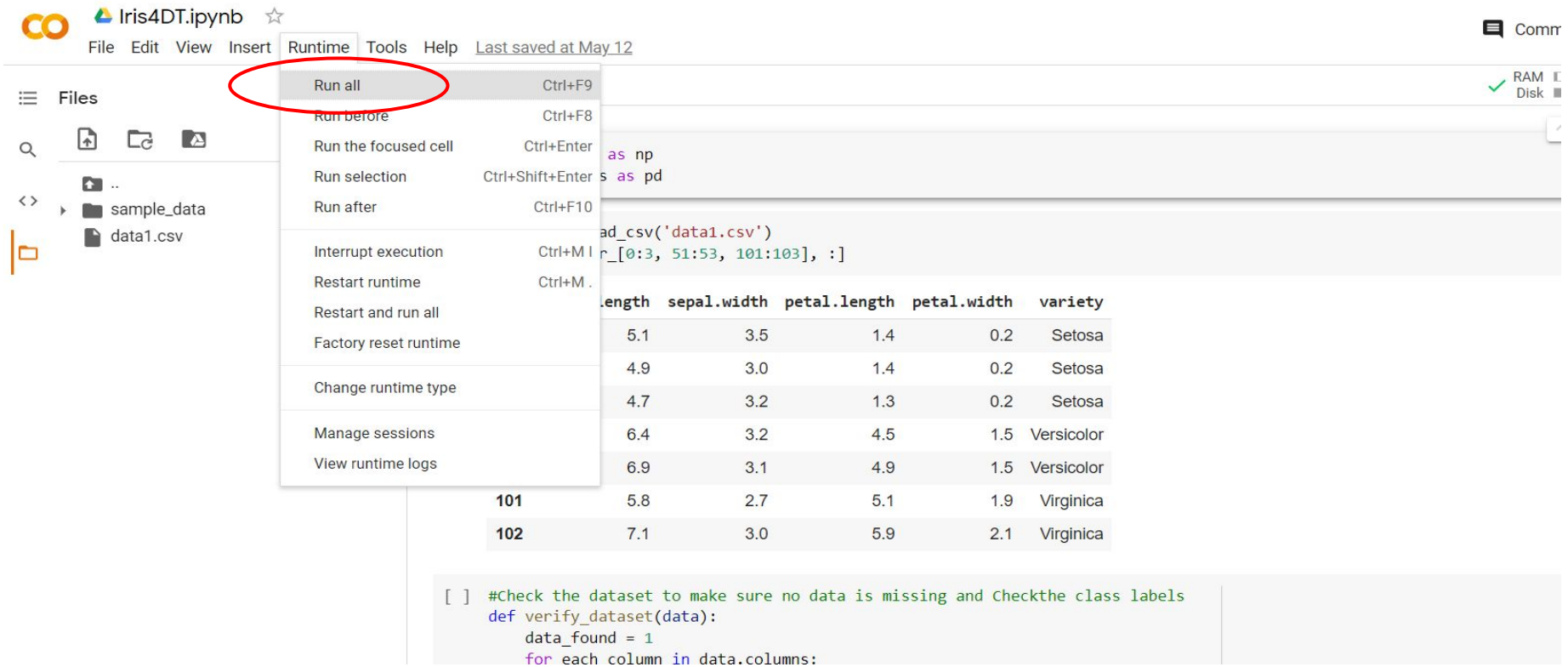
Uploading the Data Files *cont...*

- **Your files will be displayed in the colab root directory and will be read by the code**



Runtime – Run All

- To save time, you can also click ‘Runtime’ ‘Run all’ to run all the cell’s



NumPy

- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays
- NumPy is very useful in ML applications, especially with ANNs or DNNs

Pandas

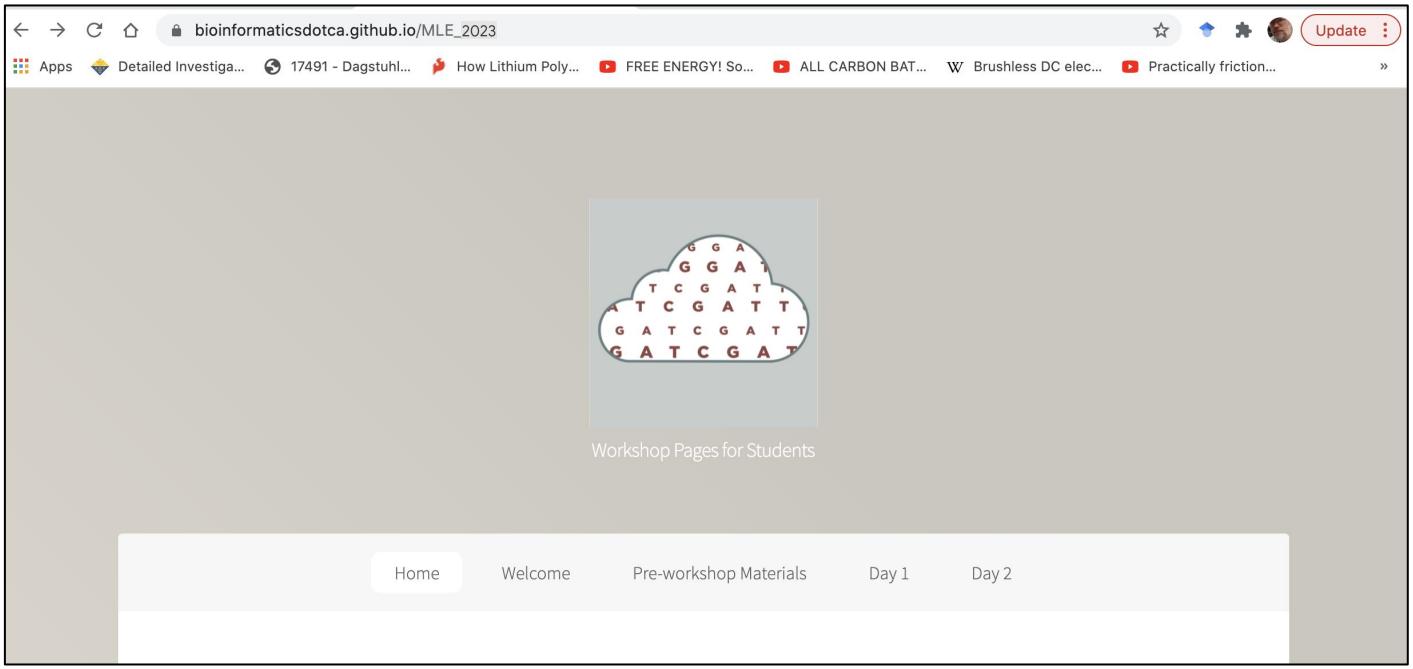
- Pandas is a Python library used for data manipulation and analysis
- it is very useful for importing data (e.g., data is in csv, txt)
- It can convert imported data into a data frame, so you have rows and columns and has widespread utility

A Note About R

- The R programming language can also be employed for ML
- All coding examples discussed in this course use the Python language – but equivalent R code is now available
- R can be run in Google Colaboratory, with the popular software RStudio, or via command line
- Run times may be quite different between the 2 programming languages
- To start a new colab notebook in R you must use the URL https://colab.to/r

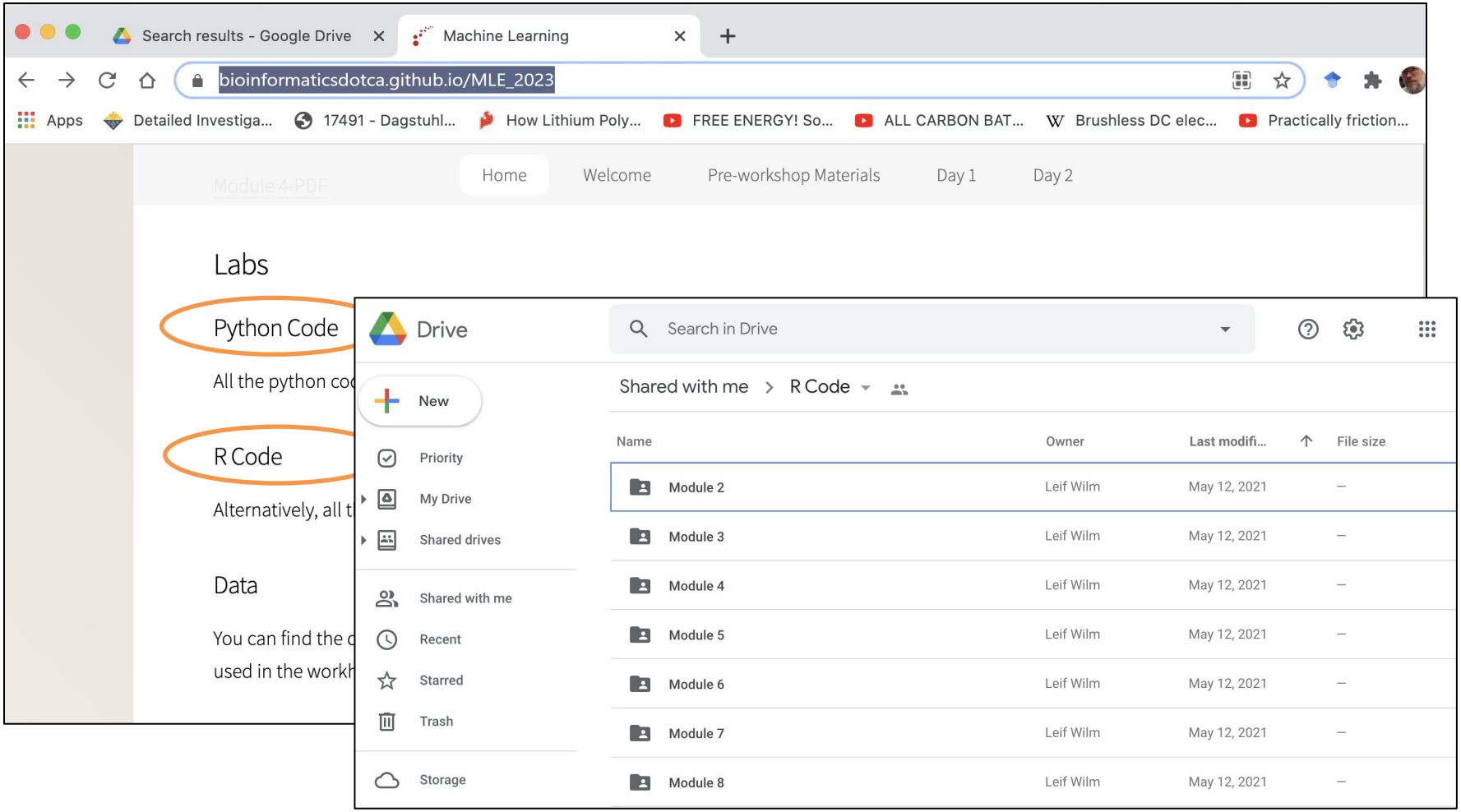
Course Code Repository

- **The course code, slides, data repository is on the GitHub [CBW Student Pages]**



https://bioinformaticsdotca.github.io/MLE_2023

Course Code Repository



ML – Anyone Can Do It

- **Scikit-learn**
  - Popular and large python library for ML programming
- **Keras**
  - Easy-to-use interface for running deep learning applications
- **Google's TensorFlow**
  - Drag & drop ML programming tool, works with Keras
- **Microsoft's Azure ML**
  - Drag and drop system for ML programming & testing
- **Facebook's Torch/PyTorch**
  - Lower-level API, but very powerful ML, gaining popularity
- **Weka/MOA**
  - Easy-to-use GUI, runs on any platform, "original" ML tool

Conclusions

- ML is a method to develop programs or algorithms automatically
- ML is great for pattern finding, data fitting and prediction – especially with large data sets
- ML comes in many different forms (ANNs, HMMs, DTs, RFs, SVMs, etc.)
- ML is used widely in everyday apps
- ML is becoming more accessible – E2U

Conclusions *cont...*

- ML has a long history of use in bioinformatics but a shorter period of use in medicine or medical genetics
- Deep learning is the hottest area of ML and is yielding surprising results and is being used in unexpected applications
- Many new ML and DL applications are starting to appear in genomics
- Often data access or ideas are the only limitations to finding new ML apps

ML is NOT the Only Game in Town

- Many (multivariate) statistical techniques can be used instead of ML techniques
- Principal Component Analysis (PCA) and Hierarchical clustering (HCA) can perform unsupervised clustering
- PLS-DA can perform supervised classification as well as some ANNs
- Linear/Logistic regression can perform just as well as SVM regression

We are on a Coffee Break & Networking Session

Workshop Sponsors: