Machine Learning Hackathon Workshop

Eitan Farchi
Orna Raz
Marcel Zalmanovici

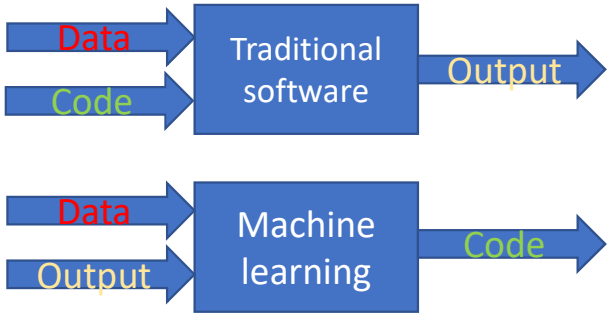
IBM Research, Haifa

Plan

- Intuitions
- Starting point for online learning
  - Data camp
  - Coursera
  - Udacity
  - Endless youtubes… Zedstatistics… DeepLizard
  - Code: stack overflow, stack exchange, sklearn documentation, …
  - Data and code: Kaggle and its kernels, UCI, …
- Things you're not told
- Hands on
  - Training models (Python, sklearn)
- Your ideas for using ML

What is ML?

- Pedro Domingos
- If you remember one thing:



What is ML?

- Peter Norwig
- If you remember one thing

> Machine learning changes the way you think about a problem:
>
> We're making observations about an uncertain world
>
> Running experiments and using statistics, not logic, to analyze the results of the experiment

What is ML?

- Supervised vs. unsupervised learning
- Other: reinforcement, online, ...
- ML model families
  - Regression
  - Classification
  - Clustering
- Supervised learning
  Andrew Ng
- Key elements of ML
  Carrie Anne

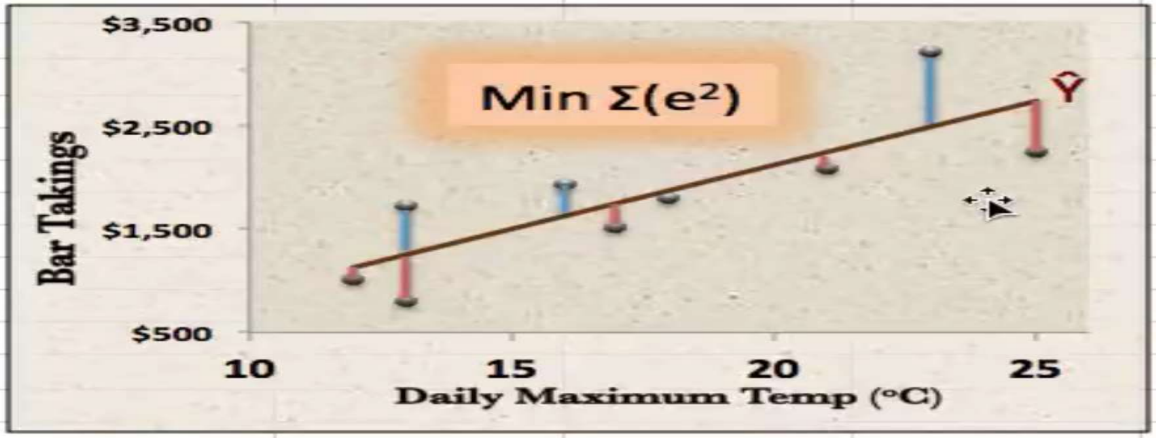
ML is regression on steroids

- Are you familiar with regression (0-5.59)? One way to view Ml is regression on steroids...which mean a harder optimization problem (one that does not have a close analytic solution and/or is not convex) with many parameters.

Day	Takings	Temp (°C)
3-Jun	$3,213	23
10-Jun	$2,089	21
17-Jun	$2,253	25
24-Jun	$1,801	18
1-Jul	$801	13
8-Jul	$1,934	16
15-Jul	$1,720	13
22-Jul	$1,514	17
29-Jul	$1,017	12

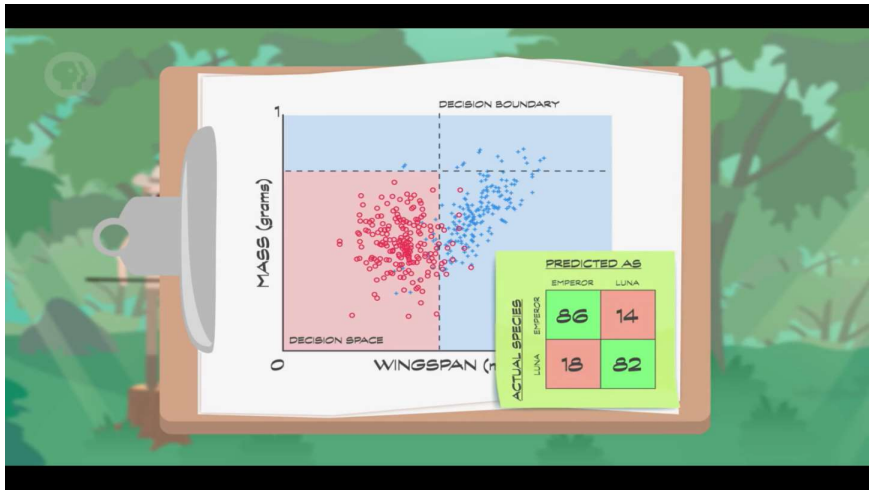
SAMPLE REGRESSION LINE

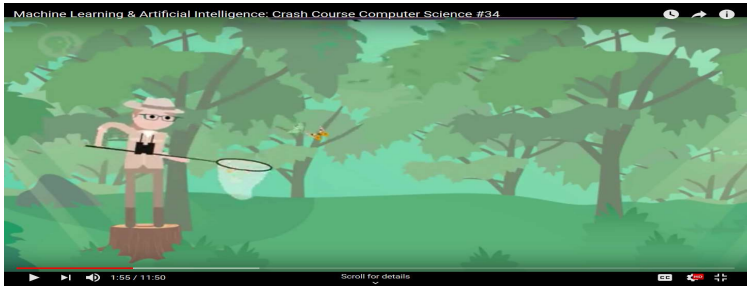
$\hat{Y} = -353.11 + 123.54X$



The problem

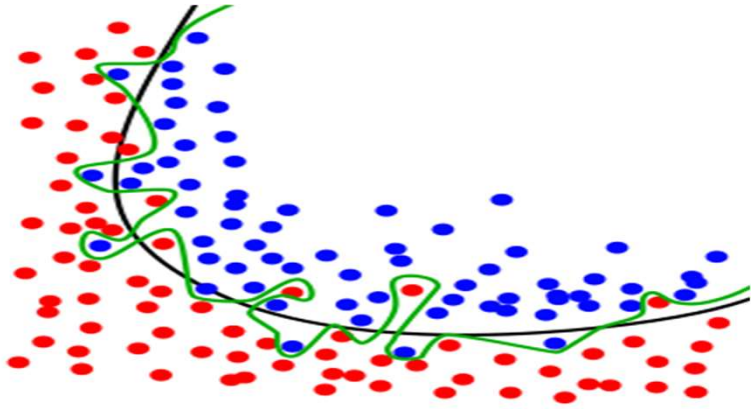
- Let's consider supervised learning first
  - You are given n labeled data points, (x1,y1),...,(xn,yn)
  - Your objective is to find a function f(x)=y that **best predicts y on a new batch of x's.**
  - When y is continuous it is called regression and when its discrete it is called classification.

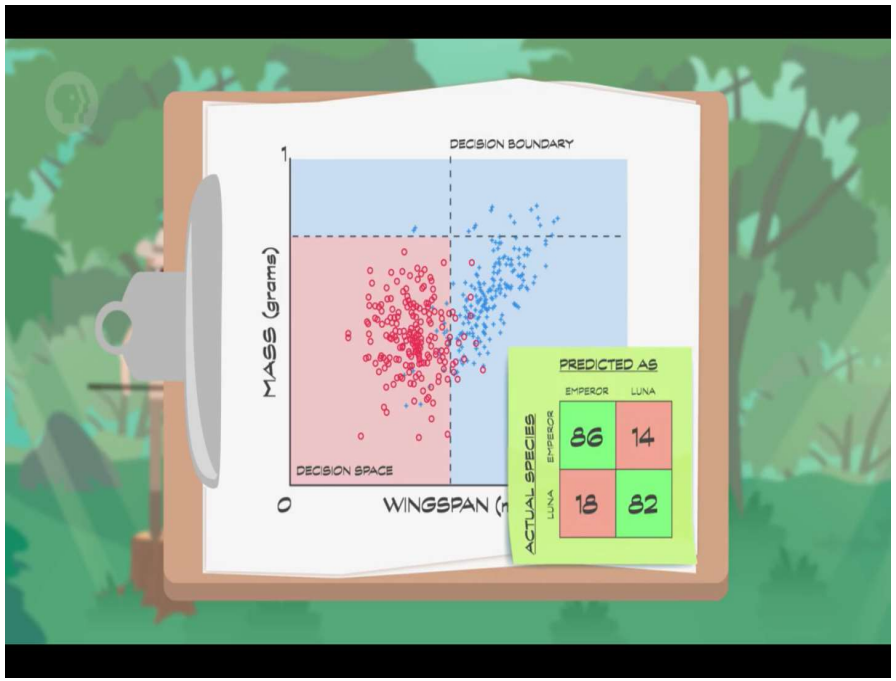




Notice that…

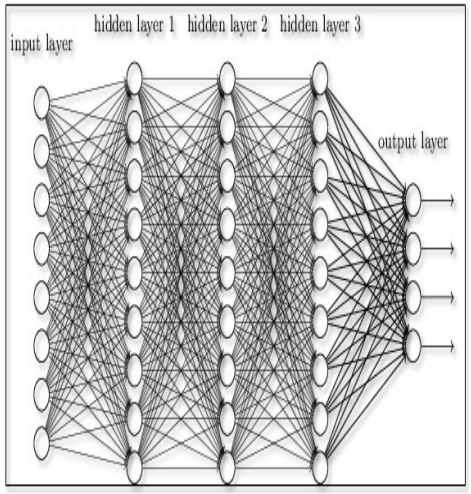
- To solve this an optimization problem is defined, e.g., a minimization of square error in our original regression problem
- Trying to explain the given data completely which is sometimes called extrapolation is a pitfall
  - You may capture random trends and your prediction power may be hindered
  - This is called overfitting

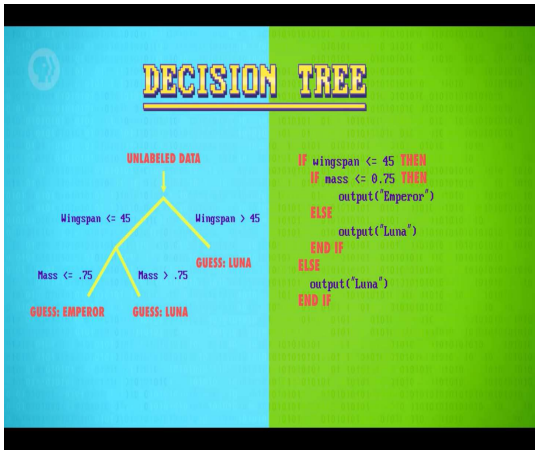


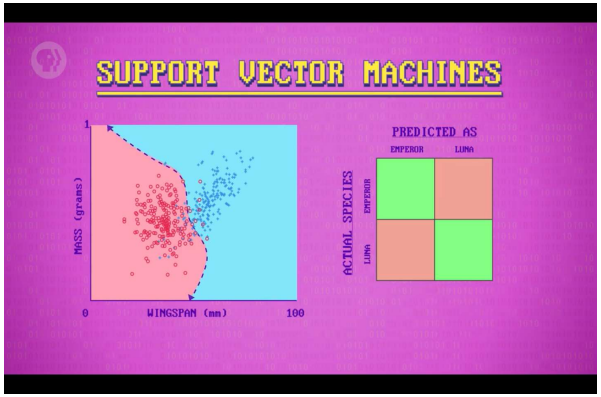


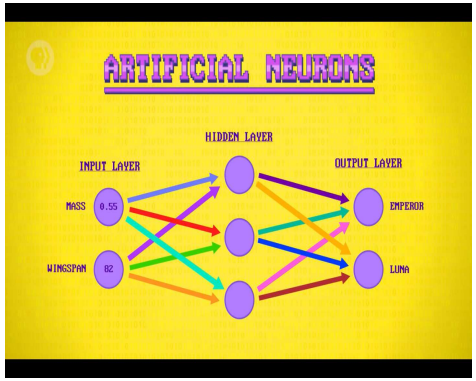
The generic solution that does not work…

- Basic intuition underlying many approaches to the classification problem
- Had we known p(x, y) and given a new x we would have calculated p(x, y) for each y and choose y with the greatest probability
- The difficulty is that it is not easy to estimate p(x, y)
- In addition there may be hundreds and thousands of features



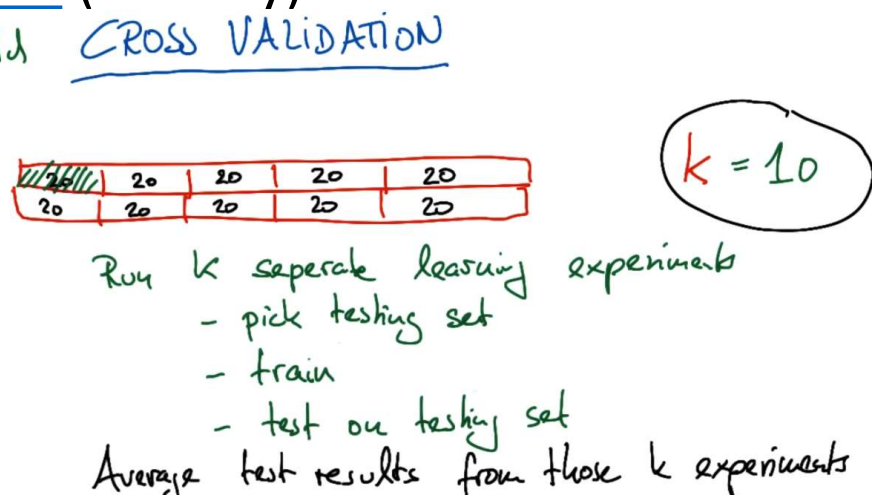






When training a ML model: Make sure you have a test set

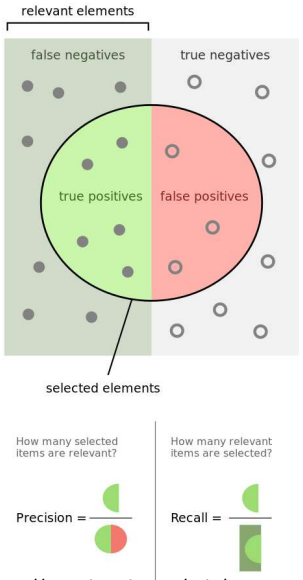
- Divide the data into training set and test set (70/30, 80/20)
- You can only assess generalization ability over the test set
- When training: Use k-fold cross validation (Udacity)
  - To avoid over-fitting (DeepLizard)
- Then assess your resulting model on the test set
  - Beware of averages
  - Use at least confidence intervals



We will use the following metrics in our handson

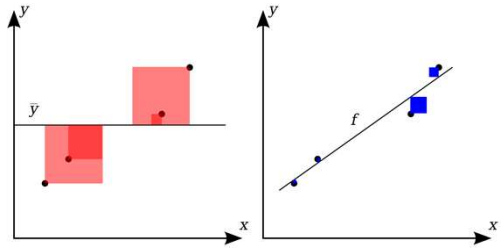
Sklearn documentation on [scoring]

- Classification: f1-weighted
  - A single score for multi-label classification that takes into account precision/recall of all classes



https://en.wikipedia.org/wiki/F1_score

- Regression: r^2
  - Proportion of the variance in the dependent variable that is predictable from the independent variable(s)

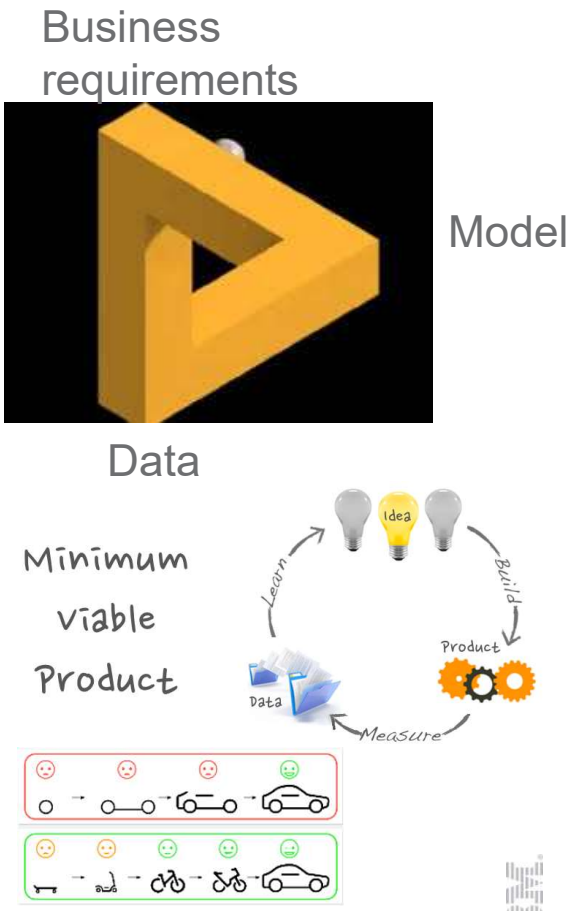


The better the linear regression (on the right) fits the data in comparison to the simple average (on the left graph), the closer the value of is to 1. The areas of the blue squares represent the squared residuals with respect to the linear regression. The areas of the red squares represent the squared residuals with respect to the average value.

https://en.wikipedia.org/wiki/Coefficient_of_determination

Is the data relevant for the business objective?

- Design time - Marry business requirements with relevant data that can be successfully learned through appropriate optimization objectives
- Define potential business objective with SME (Subject Matter Expert), data scientist and architect
  - MVP (Minimum Viable Product) approach: small iterations to both define business value and understand what value we can get
- Decision of objective depends also on the data!
  - Data may not include significant signal supporting the objective
  - Ex. Classification of software defects. ODC (Orthogonal Defect Classification) vs. custom classification
  - Data may have a signal that is ‘too strong’
  - Ex. It is always the case that a defect opened by a manager is of highest importance. No sense in learning this. Use a rule instead



Is data adequate for learning? A hybrid approach

- Identify areas of data with sufficient data for learning and areas with insufficient data for learning (see how to estimate data volume)
  - Ex. Some labels may only have a couple of examples
- Make this a design point: be able to quickly communicate to your client where there is insufficient data for learning
- What can you do with insufficient data for learning?
  - Try to create sufficient data: Over sampling? Synthetic data? Aggregating multiple labels into a single meta-label to get sufficient data?
    - Ex. If you have a labeling hierarchy, can you go up the hierarchy?
    - Ex. When you have any other structure, such as DB table indices or file system directories, can you aggregate according to this structure?
    - Consider aggregating all insufficient data into a single label 'other'. Then use a non-learning approach for that data
  - Try to provide a rough estimate solution
    - Ex. If you need to classify free-text terms into products, and you only have 1 example per product, consider a dictionary approach plus some Natural Language Processing (NLP) for new texts. If none of the dictionary terms fit, notify the user, consider asking the user to label (select a category) the data

Data cleansing

- Realistically assume that data is **always** imperfect
  - Whether automatically collected or manually entered
    - Ex. Customer data, sensor data
- Imperfection is diverse
  - Contradicting samples
  - Missing data
  - Outliers/anomalies
  - Noise of all colors
- Consider the goal of the analysis
  - Anomaly detection? Classification? …
- Consider the percentage of imperfection
  - Small? You may decide to remove/ignore imperfect data
  - Large? You may decide on a default value. Choose one answer consistently, use a different algorithm, …
- Whatever you choose to do, e.g., discard records with missing values or fill in the missing values with the average, stick with it

Create a baseline to compare your model with

- Identify a naïve, potentially no-learning approach that is easy to implement
- Always compare your current version performance to that of your baseline
- Your model should be superior to the baseline
  - The baseline serves as a sanity check/reference model
  - Can also be utilized to mitigate the existence of limited labeled data
- Are there areas where it is not better?
  - Improve your model
  - Consider a hybrid approach
  - Is your baseline good enough?
  - Simplicity has advantages
- Ex. Regression test selection. We want to execute a subset of the tests that will find all the defects in the shortest time. We can simply select the tests that found the most defects in the past and run per past run time

ID	Defects	Duration
1	0	1h
1	1	10h
2	10	1h
…		

Hands on guidance

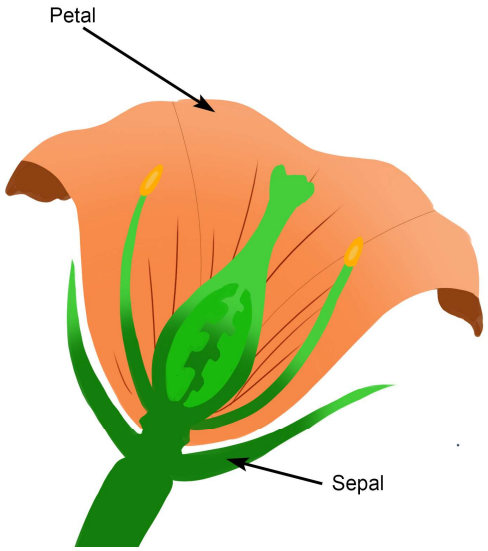
Iris dataset

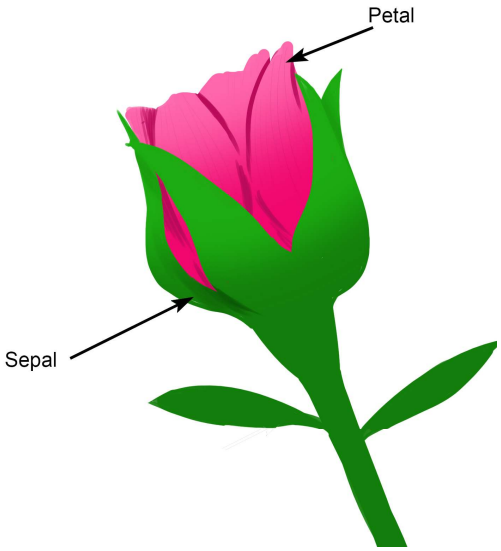
Iris-versicolor

Iris-setosa

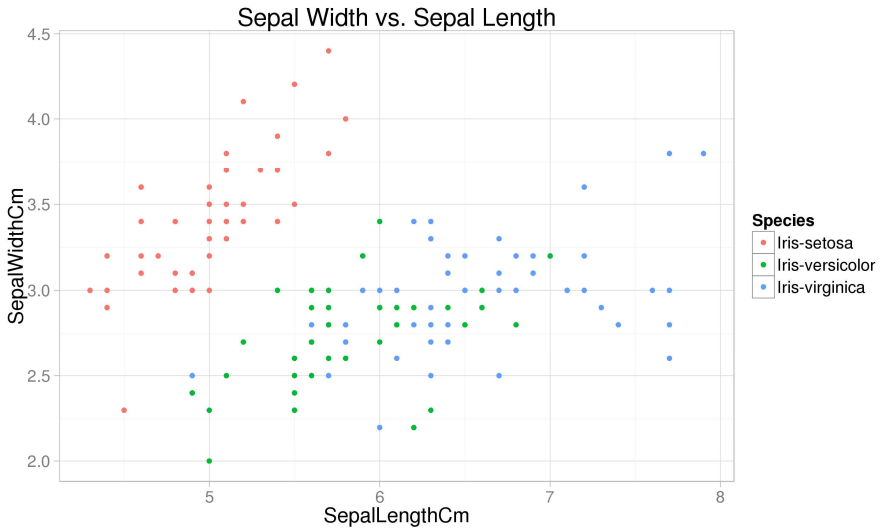
Iris-virginica

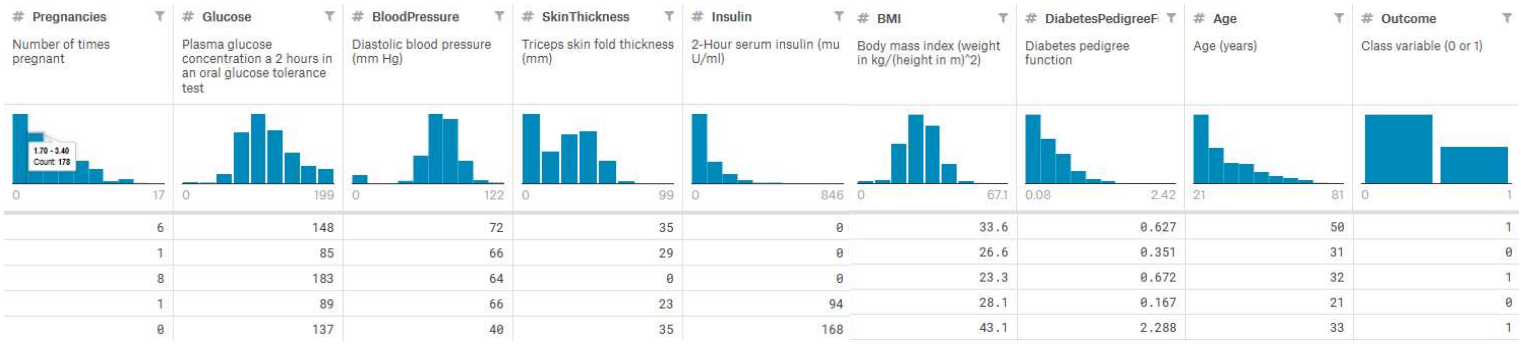
https://www.kaggle.com/uciml/iris





# Id	# SepalLengthCm	# SepalWidthCm	# PetalLengthCm	# PetalWidthCm	Species
SPL-SPW-PTL-PTW(CM)	Length of the sepal (in cm)	Width of the sepal (in cm)	Length of the petal (in cm)	Width of the petal (in cm)	Species name
1 150	4.3 7.9	2 4.4	1 6.9	0.1 2.5	Iris-setosa Iris-versicolor Other (1)
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa





# Pregnancies	# Glucose	# BloodPressure	# SkinThickness	# Insulin	# BMI	# DiabetesPedigreeF	# Age	# Outcome
Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable (0 or 1)
1.70 - 3.40 Count: 178								
0 17	0 199	0 122	0 99	0 846	0 67.1	0.08 2.42	21 81	0 1
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Diabetes dataset

https://www.kaggle.com/naveenkhasa/pima-indian-diabetes-dataset/data

sklearn models

- Install if needed
  - Anaconda 3 /python 3.x
  - Pycharm community /eclipse / spyder
  - pip install sklearn
  - pip install pandas
    (requires NumPy, SciPy, and matplotlib)
- Documentation
  - https://scikit-learn.org/stable/index.html
- Hands-on
  - TODO1: Load CSV data
  - TODO2: Instantiate several ML models
  - TODO3: Run models on data using k-fold cross-validation

Hands-on results

	Acc mean	Low	High
Linear SVM	0.721	0.656	0.786
Radial SVM	0.649	0.545	0.721
Logistic Regression	0.731	0.652	0.812
KNN	0.701	0.630	0.766
Decision Tree	0.701	0.630	0.763

Important things to note

- CI width -- ML model stability -- is important
  E.g. Linear SVM is better than Logistic Reg. in that regard
- Except Radial SVM which is worst than the others, it is hard to pick up the best solution (mean is not everything)

Your ideas for using ML

- Template
  - Data to be used + sample
  - ML technique to be applied
    - Supervised, Unsupervised
    - Assuming supervised: regression, classification
  - What will the model do
  - What will be the business value

Example:
Choose the right HyperParameter (hp)
Data:
Success rate of each hp
What will the model do :
Find the best hp combination,
ML technique :
Search problem -> Genetic
What will be the business value:
Improve performance of an ML model
by optimizing its configurations

Your example

- **Data to be used**
  -
- **ML technique to be applied**
- **What will the model do**
- **What will be the business value**

ML model review questions

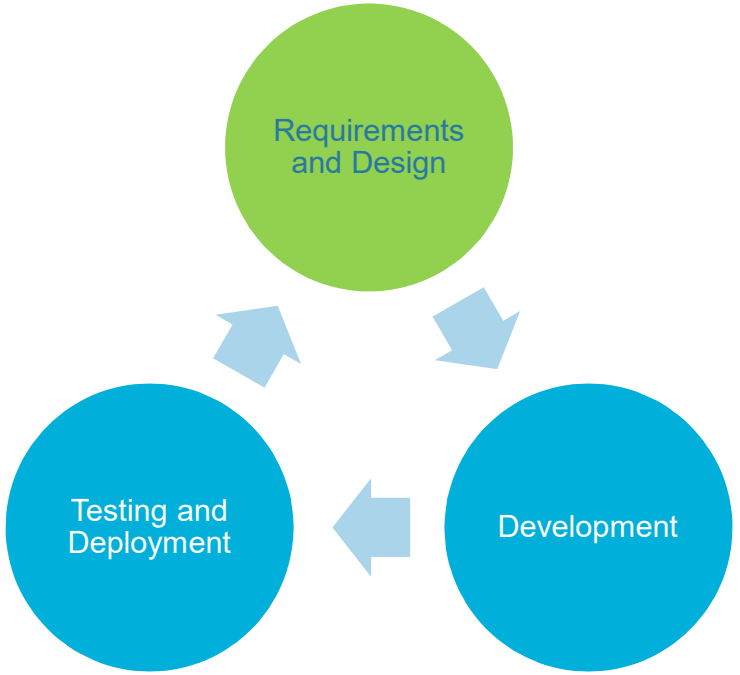
1. Enough labeled data compared to the number of features.
2. Is there a baseline that I compare my model against -- either a non learning approach or a simple model? how does the model compare with the baseline?
3. Model is not too complex (to many parameters).
4. Can I obtain additional data? Through tracing or otherwise
5. Did I loose information by my data transformation?
6. Is the entire data pipeline automated, starting from raw data?
7. Did I use a test set that was never used for training to estimate the model performance?
8. Can I generate labeled data for my problem? E.g., implement a workload to obtain server performance
9. Is the experiment by which I calculate the system performance the right one? Does it represent the desired business value?
10. Do I have hard coded hyper parameters? Can they be avoided?
11. Is there any reason to believe the model is overfitting?
12. Am I using confidence intervals to assess the ML performance of my models and not just averages? Am I training using k-fold cross-validation?
13. Can I identify important data areas (slices) that my model should handle? Did I test for the model performance over thes? (e.g.,, different test environments or test stages)

Data pipeline reparability and automation checklist

- For every result, keep track of how it was produced
- Avoid manual data manipulation steps
- Archive the exact versions of all external programs used
- Version control all custom scripts
- Record all intermediate results, when possible in standardized formats
- For analyses that include randomness, note underlying random seeds
- Always store raw data behind plots
- Generate hierarchical analysis output, allowing layers of increasing detail to be inspected
- Connect textual statements to underlying results
- Provide public access to scripts, runs, and results

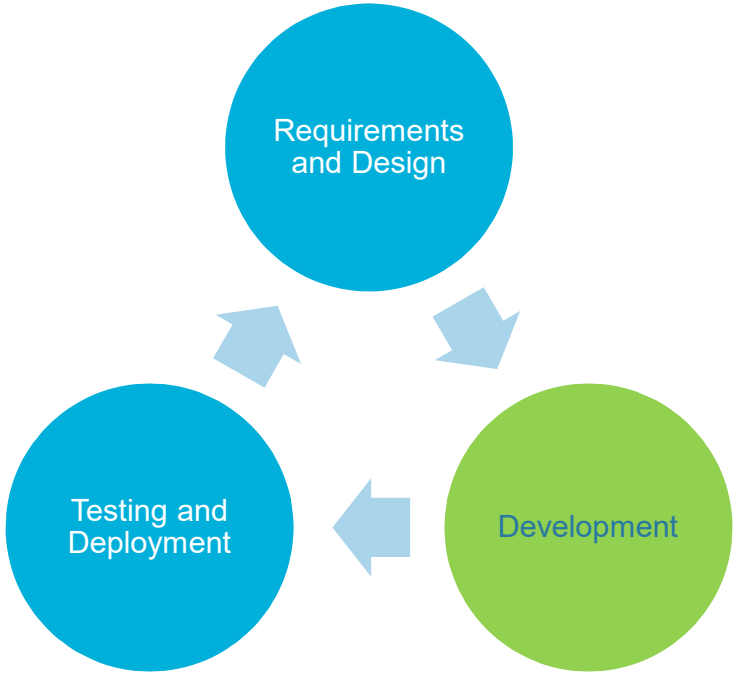
ML checklist
Requirements and design

- System business value metric is defined and a test set not used in learning is obtained
- Training data represents real life data
- There is 10 times more data than features if a generalization model is to be obtained
- Per-labels there are at least 30 examples
- Labeling accuracy is statistically estimated
- On going labeling is possible



ML checklist
Development

- Relearning from raw data is automated (the entire data pipe line is automated)
- For an object, a mapping that obtains its characteristic for each stage of the pipeline exists
- A complete list of hyper parameters is available. For each hyper parameter determine if its optimal value was searched for automatically as part of the automatic data pipeline. Each hyper parameter that was not searched for increases the risk
- Model was developed using advanced statistical techniques that estimate its performance (at least k-fold cross validation)
- Non parametric confidence intervals should be developed for all averages
- Statistical assumption made during the data pipeline development are made explicit and an appropriate check is implemented at deployment time. E.g., two features are correlated and one of them is dropped. At deployment time we check if the two features are still correlated



ML checklist
Testing and deployment

- Testing was done on the ground truth that was never used for learning using advanced statistical techniques.
- At least boot strapping and non-parametric confidence intervals
- Ground truth is sliced to identify weakness in the system
- Statistical assumptions developed at the development stage are being checked and appropriate alerts are in place if they fail
- On going labeling is implemented