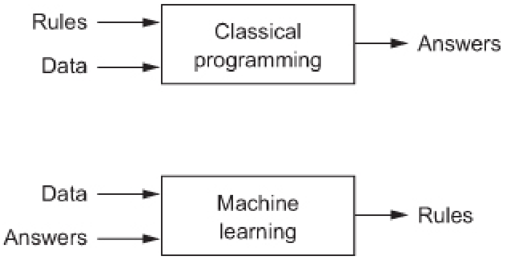
Building a Robot Judge: Data Science for the Law

2. Machine Learning Essentials

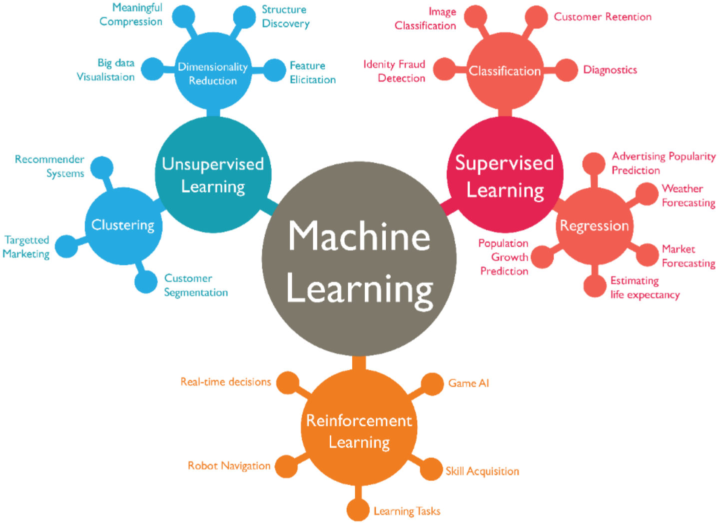
Elliott Ash

What is machine learning?



- In classical computer programming, humans input the rules and the data, and the computer provides answers.
- In machine learning, humans input the data and the answers, and the computer learns the rules.

The Machine Learning Landscape



The statistical problem

- We have a corpus of evidence documents, D, whose features can be represented as a big matrix X.
- The outcome or label to predict, Y, is the judge decision (say guilty or innocent).
- It is a function
$$Y = h(X, \epsilon)$$
of the evidence X and other factors ϵ:
- What can we learn about $h(\cdot)$?

Constructing X

- First, we will work on transforming a corpus D into a matrix of features X:
- Featurization:
  - removal of uninformative content, such as capitalization and punctuation
  - frequency counts over words and phrases
  - extraction of syntactic relations (e.g. “defendant is male and 24 years old”)

Understanding X

- The second question is how to understand the predictors X, which is an unwieldy high-dimensional object.
  - Normal descriptive methods for low-dimensional data do not work.
- Unsupervised learning and dimension reduction:
  - document similarity and clustering
  - topic model
- Supervised dimension reduction:
  - feature selection (removal of weak predictors)
  - structural topic model

Predicting $h(X)$

- The third task is to predict an outcome Y given X, that is, constructing an approximation of $h(X)$.
  - With high-dimensionality and multi-collinearity, normal regression methods do not work.
- Supervised learning:
  - regularized regression, random forests, neural nets, etc.
- In particular, need to form approximations of $h(\cdot)$ that generalize to held-out data:
  - cross-validation.

Causal estimates for $h(X)$ parameters

- Consider the linear model

$$Y_i = \alpha + X_i'\beta + A_i + \epsilon_i$$

  - A_i is an unobserved confounder: $\mathbb{E}(X_i A_i) \neq 0$
  - we have omitted variable bias; least-squares estimates for β are biased.
  - exogenously changing some of the evidence X will not have the estimated effect β on the outcome.
- Toward the end of the course we explore methods for causal inference in high dimensions:
  - orthogonalized machine learning
  - regularized instrumental variables
  - deconfounding

Setting up Python and Jupyter

- Instructions for setting up Python, as well as links to all of the code examples, are linked from the syllabus.
- Course demonstrations will be done (and problem sets should be submitted) as Jupyter notebooks
  - see Geron, Chapter 2.
  - Navigate to your directory, and at terminal, type "jupyter notebook"
  - open a browser (if it doesnt open automatically) and navigate to `http://localhost:8888/`
  - Click "New..." then "Python 3" to start a new notebook.
- Dr. Labzina will help you get set up at the first code lecture, which starts this week.

A Machine Learning Project, End-to-End

Aurelien Geron, *Hands-on machine learning with Scikit-Learn & TensorFlow*, Chapter 2:

1. Look at the big picture.
2. Get the data.
3. Discover and visualize the data to gain insights.
4. Prepare the data for Machine Learning algorithms.
5. Select a model and train it.
6. Fine-tune your model.
7. Present your solution.
8. Launch, monitor, and maintain your system.

Select a performance measure

- A typical performance measure for regression problems is Mean Squared Error (MSE):

$$\text{MSE}(X, h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2$$

  - m, the number of rows/observations
  - X, the feature set, with row x_i
  - Y, the outcome, with item y_i
  - $h(x_i)$ the model prediction (hypothesis)
- Econometricians are familiar with MSE because it's the cost function for the OLS estimator.
  - it corresponds to the Euclidian norm or L2 norm, notated as $||\cdot||_2$

Other cost functions

- Mean Absolute Error:

$$\text{MAE}(X, h) = \frac{1}{m}\sum_{i=1}^{m} |h(x_i) - y_i|$$

  which corresponds to the L1 norm (or in econometrics, quantile regression).
  - the L1 norm is less sensitive to outliers than the L2 norm.
- MSE is a good baseline, but other cost functions are sometimes used.
  - for classification, start with F1 score.
  - for regression, start with R^2.

Training and Testing

- Machine learning models can achieve arbitrarily high accuracy in-sample.
  - performance should be evaluated out-of-sample.
- simplest approach:
  - randomly sample a training dataset for learning parameters
  - form predictions in testing dataset for evaluating performance.

Data Prep for Machine Learning

- See Geron Chapter 2, pp. 61-68 for pandas and sklearn syntax:
  - impute missing values.
  - feature scaling (often helpful/necessary for ML models to work well)
  - encode categorical variables.
- Best practice: reproducible data pipeline.

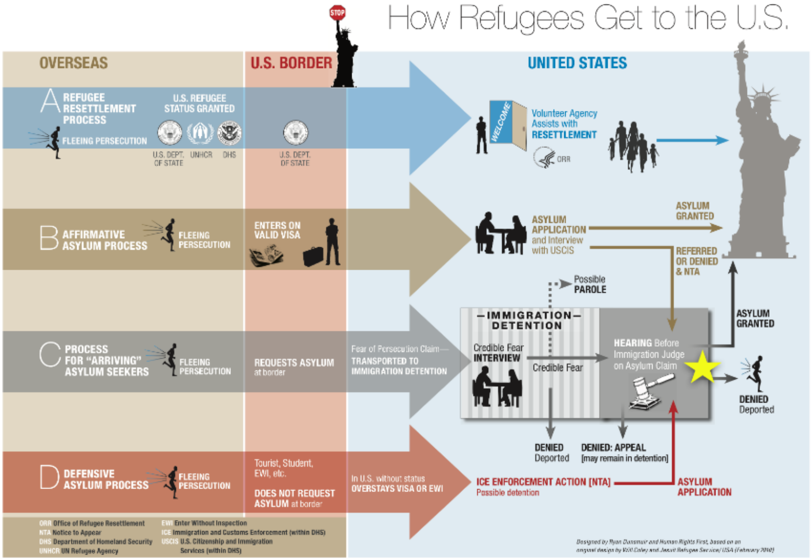
Scikit-Learn Design Principles

- **Consistency:**
  - *Estimator:* An object that can estimate parameters. Estimation is performed by `fit()` method. Exogenous parameters (provided by the researcher) are called *hyperparameters*.
  - *Transformer*: An object that transforms a data set. Transformation is performed by the `transform()` method. The convenience method `fit_transform()` both fits an estimator and returns the transformed input data set.
  - *Predictor*: An object that forms a prediction from an input data set. The `predict()` method forms the predictions. It also has a `score()` method that measures the quality of the predictions given a test set.
- **Inspection:** Hyperparameters and parameters are accessible. Learned parameters have an underscore suffix (e.g. `lin_reg.coef_`)
- **Non-proliferation of classes:** Use native Python data types; existing building blocks are used as much as possible.
- **Sensible defaults:** Provides reasonable default values for hyperparameters – easy to get a good baseline up and running.

Cross-Validation

- Use `cross_val_score` method to get model performance across subsets of data.
- Use `GridSearchCV` or `RandomizedSearchCV` to automate search over parameter space.

Asylum in the U.S.



Source: rcusa.org.

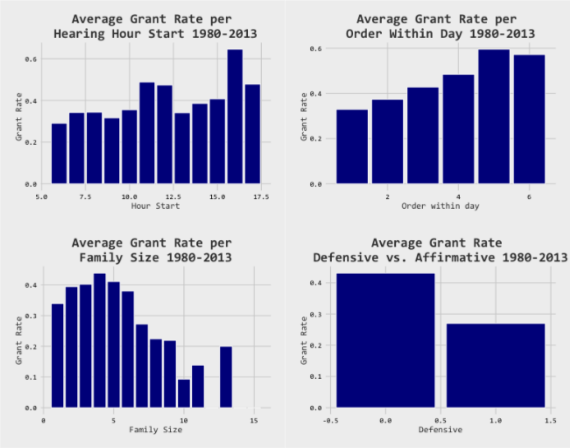
Dunn et al (2017): Asylum Courts

- Data:
  - universe of asylum court cases, 1981-2013
  - 492,903 decisions, 336 courts, 441 judges
- High stakes: denial of asylum results in deportation.
- Average grant rate: 35%.

Top Ten Countries by Applicants

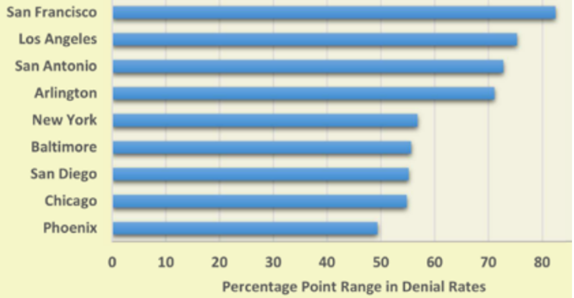
Country	**Count**	**Percentage**	**Grant Rate**
China	107964	0.19	0.53
Haiti	42013	0.074	0.16
El Salvador	41626	0.074	0.087
Guatemala	34705	0.061	0.11
Colombia	27713	0.049	0.35
India	19161	0.034	0.37
Mexico	19031	0.034	0.073
Nicaragua	15987	0.028	0.2
Albania	12036	0.021	0.52
Indonesia	11399	0.02	0.32

Variation in Grant Rates



- Judges are more lenient before lunch; U-shape in family size.

U.S. Asylum Courts: Disparities in Grant Rates



- In San Francisco, one judge grants 90.6% of asylum requests, while another judge grants just 2.9%!

Predicting U.S. Asylum Court Decisions

		Predicted	
		Denied	Granted
True	Denied	195,223	65,798
	Granted	73,269	104,406

Accuracy = 68.3%, F1 = 0.60

- Prediction App (Beta): `https://floating-lake-11821.herokuapp.com/`
  - predictions made using logistic regression with L2 regularization, penalty selected by cross-validation grid search.

Judge Variation in Predictability, Snap Judgements

- There may be cases for which country and date of application should completely determine outcomes (e.g., during violent conflict)
  - But significant inter-judge disparities in predictability suggest that this understanding of the country circumstances does not apply to all
- Some judges are highly predictable, always granting or rejecting:
  - Snap judgments and predetermined judgments (Ambady and Rosenthal 1993)
  - Stereotypes pronounced with time pressure & distraction (Bless et al 1996)
    - "In a crowded immigration court, 7 minutes to decide a family's future" (Wash Post 2/2/14

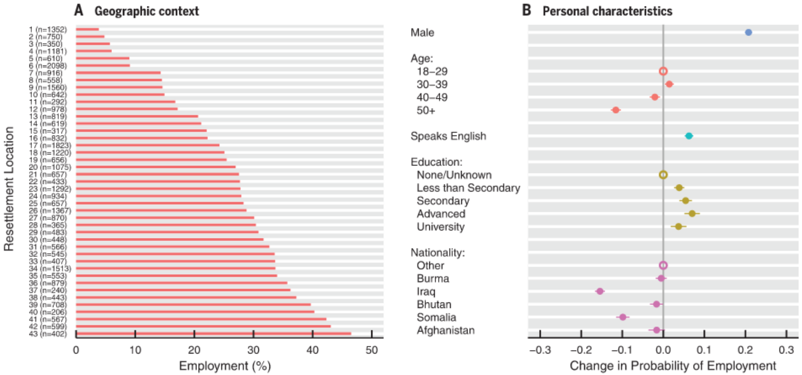
Judge Identity is Most Predictive

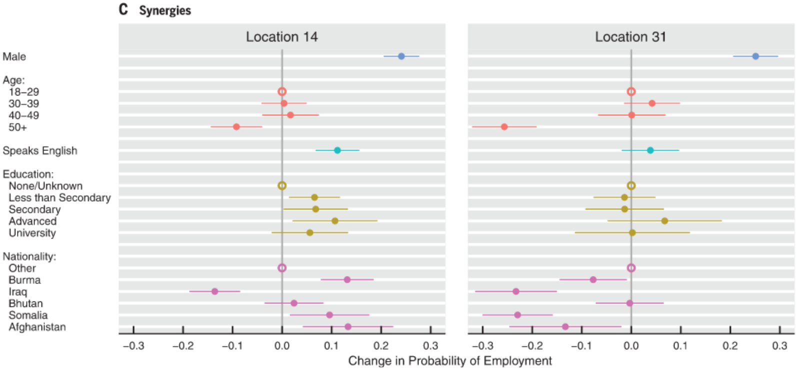
Model	Accuracy	ROC AUC
Judge ID	0.71	0.74
Judge ID & Nationality	0.76	0.82
Judge ID & Opening Date	0.73	0.77
Judge ID & Nationality & Opening Date	0.78	0.84
Full model at case completion	0.82	0.88

- Predictions from random forest classifier, with parameters selected by cross-validated grid search.
  - Training/test split 482K/120K.

Bansak, Ferwerda, Hainmueller, Dillon, Hangartner, Lawrence, and Weinstein (2018)

- This paper develops algorithm for assigning refugees across resettlement locations:
  - predict refugee employment rate using location and refugee characteristics
  - uses combination of supervised machine learning and optimal matching
  - discover synergies between refugee characteristics and resettlement sites
- Predicted gains after assignment based on optimal match:
  - Employment rate increased by 40% in United States, 75% in Switzerland.


A Geographic context
Resettlement Location
1 (n=1352)
2 (n=750)
3 (n=350)
4 (n=1181)
5 (n=610)
6 (n=2098)
7 (n=916)
8 (n=558)
9 (n=1560)
10 (n=642)
11 (n=292)
12 (n=978)
13 (n=819)
14 (n=619)
15 (n=317)
16 (n=832)
17 (n=1823)
18 (n=1220)
19 (n=656)
20 (n=1075)
21 (n=657)
22 (n=433)
23 (n=1292)
24 (n=934)
25 (n=657)
26 (n=1367)
27 (n=870)
28 (n=365)
29 (n=483)
30 (n=448)
31 (n=566)
32 (n=545)
33 (n=407)
34 (n=1513)
35 (n=553)
36 (n=879)
37 (n=240)
38 (n=443)
39 (n=708)
40 (n=206)
41 (n=567)
42 (n=599)
43 (n=402)
0
10
20
30
40
50
Employment (%)
B Personal characteristics
Male
Age:
18–29
30–39
40–49
50+
Speaks English
Education:
None/Unknown
Less than Secondary
Secondary
Advanced
University
Nationality:
Other
Burma
Iraq
Bhutan
Somalia
Afghanistan
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Change in Probability of Employment

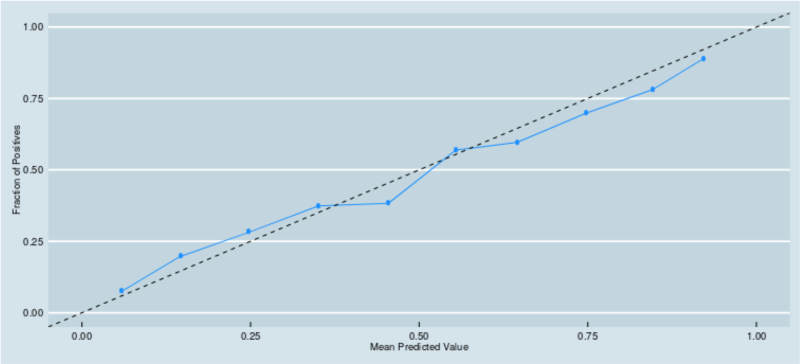

C Synergies
Location 14
Location 31
Male
Age:
18–29
30–39
40–49
50+
Speaks English
Education:
None/Unknown
Less than Secondary
Secondary
Advanced
University
Nationality:
Other
Burma
Iraq
Bhutan
Somalia
Afghanistan
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Change in Probability of Employment

Machine Learning

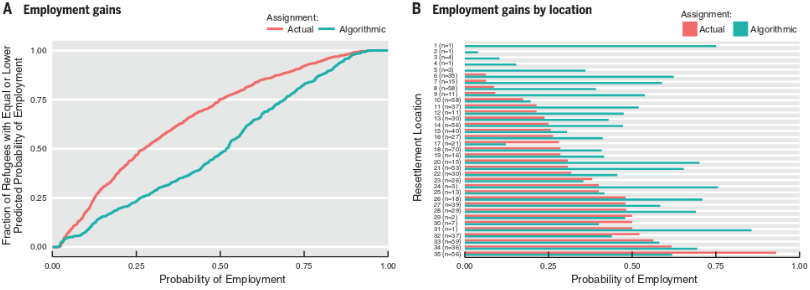
- Implement model prediction for each location:
  - predict employment given refugee characteristics
- Best model: gradient boosted trees
  - others tried: random forests, regularized logistic, kernel-based regularized least squares.
  - Model is 74% accurate for predicting employment status (compared to 66% accurate if guessing unemployed).

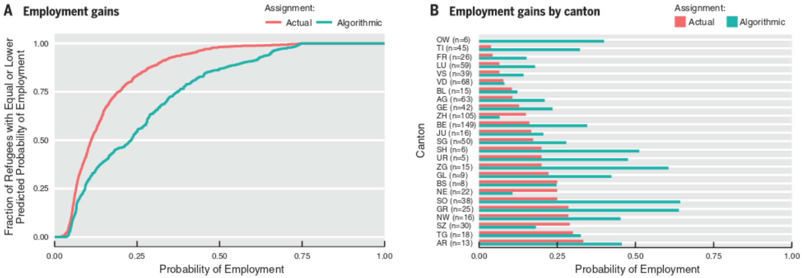
Distributional Accuracy

(a) Calibration plot



- Divide refugees into 10 bins based on **predicted probability of employment** (x-axis)
  - y-axis gives **true employment rate** for that bin of individuals.


A Employment gains
Assignment:
Actual
Algorithmic
Fraction of Refugees with Equal or Lower Predicted Probability of Employment
0.00
0.25
0.50
0.75
1.00
Probability of Employment
B Employment gains by location
Resettlement Location


A Employment gains
Assignment:
Actual
Algorithmic
Fraction of Refugees with Equal or Lower Predicted Probability of Employment
0.00
0.25
0.50
0.75
1.00
Probability of Employment
B Employment gains by canton
Canton
OW (n=6)
TI (n=45)
FR (n=26)
LU (n=59)
VS (n=39)
VD (n=68)
BL (n=15)
AG (n=63)
GE (n=42)
ZH (n=105)
BE (n=149)
JU (n=16)
SG (n=50)
SH (n=6)
UR (n=5)
ZG (n=15)
GL (n=9)
BS (n=8)
NE (n=22)
SO (n=38)
GR (n=25)
NW (n=16)
SZ (n=30)
TG (n=18)
AR (n=13)