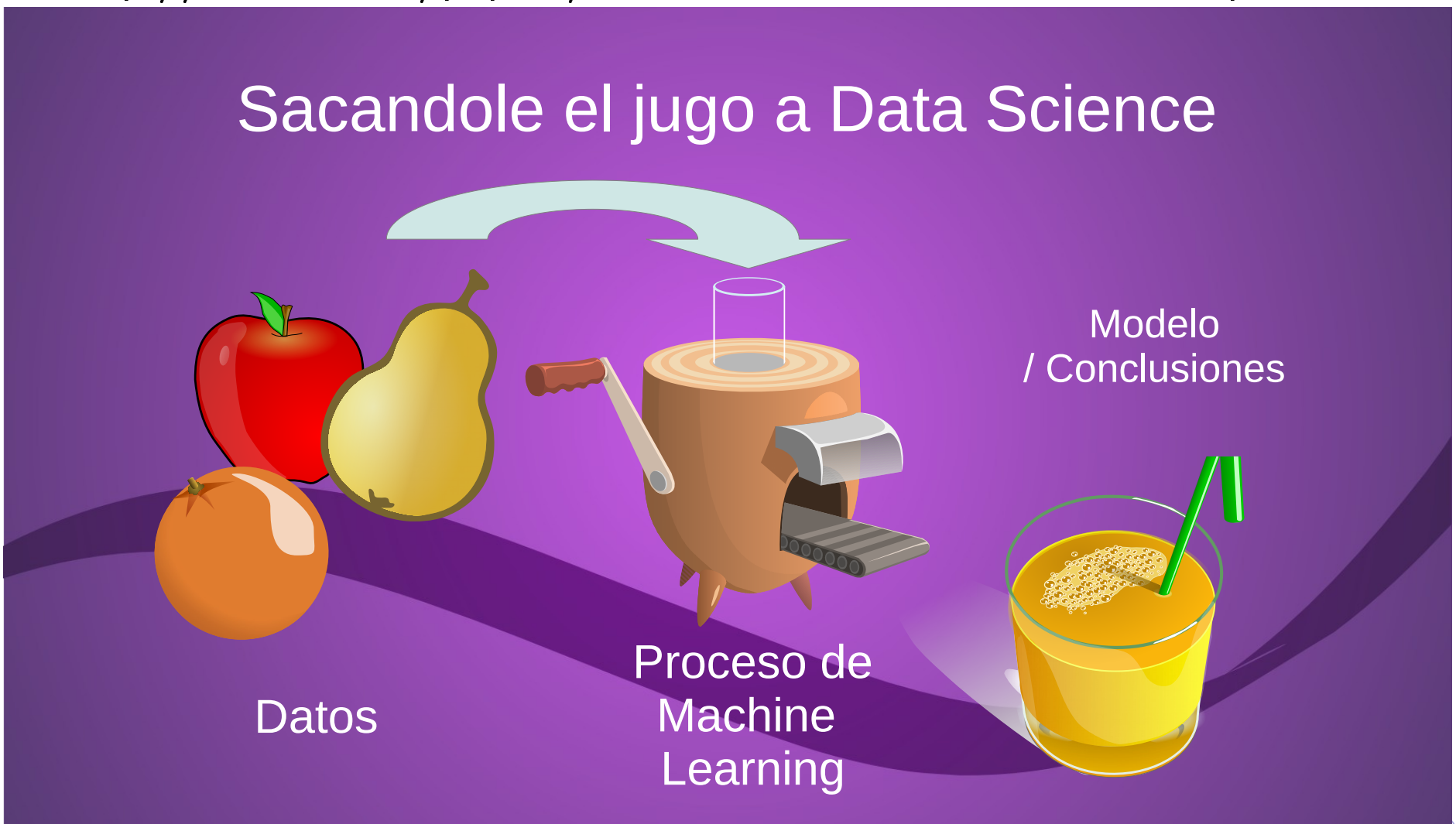
El Factor Humano en la Ciencia de Datos

Pablo Duboue, PhD

Grupo Análisis y Procesamiento de Grandes Redes Sociales y Semánticas
UNC Cordoba
Marzo 2021

De Cordoba Tech Week 2019

http://duboue.net/papers/dubouеCordobaTechWeekThu12.pdf



De Cordoba Tech Week 2019

No haciéndolo bien

- Tiramos la fruta dentro
- Analizamos la composición química del jugo
- Probamos cambios en la juguera

De Cordoba Tech Week 2019

Donde hacer cambios

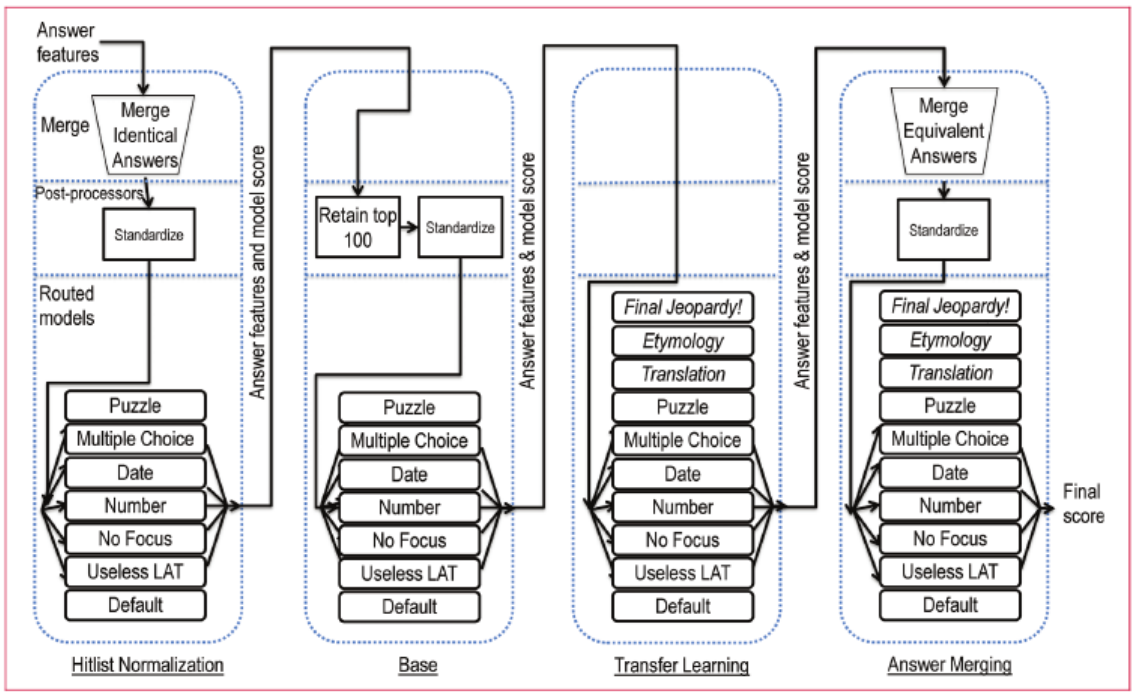
- Elegir mejor la fruta: improved data acquisition
- Elegir las mejores partes de la fruta: improved “features”
- Probar el jugo: custom error metrics
- Preguntar a expertos en jugos (no en ML):
  - Qué frutas son buenas.
  - Cuáles son las mejores partes de las frutas.
  - Qué propiedades del jugo son importantes.

Talk in Detail

1. Basic ML Cycle
   - Cross-validation
2. Human-in-the-loop Cycle
   - Race against the machine
   - AutoML
   - Improve the input
   - Improve the output
3. Feature Engineering
   - EDA
   - EA
   - Data Stagging

About the Pablo

- PhD in Computer Science, Columbia University, NY
  - Thesis on weakly supervised learning for text generation
- IBM Research Watson
  - Deep QA - Watson - Jeopardy! Show
- Wrote a book on Feature Engineering



ML Definitions

- In this lecture, I will use the following terms:
  - Supervised learning
    - Classifiers or regressors
  - *Raw data*
  - Instance
  - Target class / target value
  - Features

ML CYCLE

1. Train/test split (*Feature vectors*)
   - *Trainset*
   - *Testset*
2. Build model (*Trainset*)
   - *Trained model*
3. Evaluate(*Trained model*, *Testset*)
   - *Evaluation results*

Example: Car Hire Gratuity

- New York City taxicab rider tipping behaviour, from Chapter 6 of

  Brink, Henrik, Richards, Joseph W, and Fetherolf, Mark, Real-world machine learning (Shelter Island, NY: Manning, 2017).

- Raw data:
  - GPS location where taxi was hired, GPS location of destination, ride cost, payment type tip, plus other data (!4 million records)
- Features:
  - Radial distance from Times Square ("centre" of city)
  - Payment type
  - Target value: tip
- Evaluation metric: area under ROC curve
- Regression problem

Result

- Training a regression over this dataset produces a model that works really well
  - Too well
- It is dominated by a single feature: payment type

Problem

- Further examination shows all "payment type = cash" trips have no tips
  - They are not recorded in the data
- Dropping all the wrong data produces a real results:
  - GPS location of destination is the main predictor
  - Trips around the centre of town net no tips

Too Simplistic?

- The basic model of "data"→"model"→"evaluation" is too narrow-minded
- We will explore expansions over this model, using it as a building block

Overfitting

- Overfitting refers to the problem of memorizing the data
  - A stopped watch is accurate twice a day
- When doing ML, we care about fitting a model to existing data
  - purpose is generalization (extrapolation)
- Overfitting happens when the model follows too closely the original training sample and it **fails to generalize**.
- Always use a separate test set when training supervised learning models
  - Evaluating on the train set
    - Results that are too optimistic
    - Not representative of the behaviour on new data
- But there are other types of overfitting
  - Inacurate sample of the overall population (collecting data only in the summer, for example)
  - Overfitting the model decision or its parameters
  - Overfitting the features

Avoiding Overfitting

- Testing on held-out set once (or very few times) is a great way to avoid overfitting
  - Changing the test set over time, stagging the training/evaluation data as we will discuss at the end of the lecture
  - Every time you quiz your test set, your process gets more tuned to the sample rather than the overall population
    - Similar to statistics problem addressed by techniques like the Bonferroni correction
- Other techniques include reducing the "capacity" (parameters) of the model
  - A dumber model can learn less things about the data
  - It will work worse on training but will (hopefully) generalize better
  - For example, early stopping when training neural networks
- A common misconception is that cross-validation will not overfit but let's take a look at cross validation in detail first

Cross-Validation

- Xval is a technique to deal with small datasets in model evaluation by reducing the loss of data allocated to testing the model.
- How it works:
  1. Split the training data into N parts ("folds"), taken at random
  2. The system is then trained and tested N times:
     - for each fold, the remaining $N-1$ folds are used to train a model,
     - which is then used to predict labels or values on the selected fold.
- In certain domains care has to be taken when splitting the data so each fold contains a full view of the data (**stratified Xval**)
  - The splitting is random over sets of instances rather than over all instances.
  - For example, if training over rounds of user logs, all rows for the same user should fall in the same fold.
    - Otherwise the evaluation will not be representative of the behaviour in production.

"Losing" Data

- We use Xval due to a lingering feeling that testing data is "wasted"
  - It is not used to estimate the parameters of our model
- But keeping data aside to understand how well our trained ML model performs on production is definitely more valuable than the marginal changes to the model that data will produce:
  - if 20% more data produces drastic changes then our model is not stable
    - We do not have much of a model, really
  - We will be better served by a simpler model with fewer parameters that behaves in a stable fashion over the available data
- Better understanding of the model behaviour on unseen data can make the model more useful in the **larger context** it will be used
  - Some of its shortcomings can be addressed with non-ML data or custom programming

Example of Cross-Validation

- Prediction task to determine whether a person makes over 50K a year. (UCI "adult" dataset.)

workclass { Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked }
education { Bachelors, Some-college, 11th, HS-grad, ...}
marital-status { Married-civ-spouse, Divorced, Never-married, ... }
occupation { Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, ... }
relationship { Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried }
race { White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black }
sex { Female, Male }
native-country { United-States, Cambodia, England, Puerto-Rico, Canada, ... }
target { >50K, <=50K }

Folds

State-gov, Bachelors, Never-married, Adm-clerical, Not-in-family, White, Male, United-States, <=50K

Self-emp-not-inc, Bachelors, Married-civ-spouse, Exec-managerial, Husband, White, Male, United-States, <=50K

Private, Masters, Married-civ-spouse, Exec-managerial, Wife, White, Female, United-States, <=50K

Private, 9th, Married-spouse-absent, Other-service, Not-in-family, Black, Female, Jamaica, <=50K

Self-emp-not-inc, HS-grad, Married-civ-spouse, Exec-managerial, Husband, White, Male, United-States, >50K

Private, Masters, Never-married, Prof-specialty, Not-in-family, White, Female, United-States, >50K

Private, Bachelors, Married-civ-spouse, Exec-managerial, Husband, White, Male, United-States, >50K

Private, HS-grad, Divorced, Handlers-cleaners, Not-in-family, White, Male, United-States, <=50K

Private, 11th, Married-civ-spouse,

Private, Some-college, Married-civ-spouse, Exec-managerial, Husband, Black, Male, United-States, >50K

State-gov, Bachelors, Married-civ-spouse, Prof-specialty, Husband, Asian-Pac-Islander, Male, India, >50K

Private, Bachelors, Never-married, Adm-clerical, Own-child, White, Female, United-States, <=50K

Private, Assoc-acdm, Never-married, Sales, Not-in-family, Black, Male, United-States, <=50K

Private, Assoc-voc, Married-civ-spouse, Craft-repair, Husband, Asian-Pac-Islander, Male, ?, >50K

Private, 7th-8th, Married-civ-spouse, Transport-moving, Husband, Amer-Indian-Eskimo, Male, Mexico, <=50K

Self-emp-not-inc, HS-grad, Never-married, Farming-fishing, Own-child, White, Male, United-States, <=50K

Private, HS-grad, Never-married, Machine-op-inspct, Unmarried, White, Male, United-States, <=50K

Private, 11th, Married-civ-spouse, Sales, Husband, White, Male, United-States, <=50K

Private, Doctorate, Married-civ-spouse, Prof-specialty, Husband, White, Male, United-States, >50K

Private, HS-grad, Separated, Other-service, Unmarried, Black, Female, United-States, <=50K

Federal-gov, 9th, Married-civ-spouse, Farming-fishing, Husband, Black, Male, United-States, <=50K

Private, 11th, Married-civ-spouse, Transport-moving, Husband, White, Male, United-States, <=50K

Private, HS-grad, Divorced, Tech-support, Unmarried, White, Female, United-States, <=50K

Local-gov, Bachelors, Married-civ-spouse, Tech-support, Husband, White, Male, United-States, >50K

Private, HS-grad, Never-married, Craft-repair, Own-child, White, Male, United-States, <=50K

?, Some-college, Married-civ-spouse, ?, Husband, Asian-Pac-Islander, Male, South, >50K

Private, HS-grad, Divorced, Exec-managerial, Not-in-family, White, Male, United-States, <=50K

Private, HS-grad, Married-civ-spouse, Craft-repair, Husband,

Models

- For models, we will use ID3 decision trees as they are simple to run and tend to overfit badly on small datasets without any prunning
  - That will help highlight some of the problems with Xval

Results

Fold 1	Fold 2	Fold 3
50% acc	30% acc	acc 40%
education = Bachelors | workclass = Private: <=50K | workclass = Self-emp-not-inc: null | workclass = Self-emp-inc: null | workclass = Federal-gov: null | workclass = Local-gov: >50K | workclass = State-gov: >50K | workclass = Without-pay: null | workclass = Never-worked: null education = Some-college: >50K education = 11th: <=50K education = HS-grad: <=50K education = Prof-school: null education = Assoc-acdm: <=50K education = Assoc-voc: >50K education = 9th: <=50K education = 7th-8th: <=50K education = 12th: null education = Masters: >50K education = 1st-4th: null education = 10th: null education = Doctorate: >50K education = 5th-6th: null education = Preschool: null	education = Bachelors | workclass = Private | | occupation = Tech-support: null | | occupation = Craft-repair: null | | occupation = Other-service: null | | occupation = Sales: null | | occupation = Exec-managerial: >50K | | occupation = Prof-specialty: <=50K | | occupation = Handlers-cleaners: null | | occupation = Machine-op-inspct: null | | occupation = Adm-clerical: null | | occupation = Farming-fishing: null | | occupation = Transport-moving: null | | occupation = Priv-house-serv: null | | occupation = Protective-serv: null | | occupation = Armed-Forces: null	occupation = Tech-support: null occupation = Craft-repair: >50K occupation = Other-service: <=50K occupation = Sales: <=50K occupation = Exec-managerial | relationship = Wife: <=50K | relationship = Own-child: null | relationship = Husband | | workclass = Private: >50K | | workclass = Self-emp-not-inc | | | education = Bachelors: <=50K | | | education = Some-college: null | | | education = 11th: null | | | education = HS-grad: >50K | | | education = Prof-school: null | | | education = Assoc-acdm: null | | | education = Assoc-voc: null | | | education = 9th: null | | | education = 7th-8th: null | | | education = 12th: null | | | education = Masters: null | | | education = 1st-4th: null | | | education = 10th: null | | | education = Doctorate: null | | | education = 5th-6th: null

Cross-validation Problems

- Cross Validation is not without issues, though.
- We will discuss three problems with Xval:
  - Multiple models
  - Multiple results
  - Unevaluated final model

Multiple Models

- Each fold trains a different model
- That makes error analysis very difficult, because the models might potentially be very different
  - In particular, if our model has too many parameters and our data is too small
  - Which is what bootstrapping helps detect: stability issues

Multiple Results

- Average the evaluation metrics per fold: macro evaluation
- Use the labels (obtained from different models) over the whole dataset: micro evaluation
- Usually the two results will disagree, sometimes very strongly
  - Again, Xval is a particular case of bootstrapping, which is great to evaluate the **stability** of a solution
  - Variance of the macro evaluation is a good estimator of the stability of the model over the available data

Final model is unevaluated

- With Xval, we train as many models as folds. Which model to deploy?
- Usually, a final model is trained on **all data**
- But this model has not been evaluated
- If all the data together triggers a patological condition in the code (i.e., all parameters go to zero) **we won't know**
  - This is highly unlikely, but more common situations (i.e., corrupting the data before final model training) that result in an unusable model will go unchecked

Race Against the Machine

- Race Against the Machine is a short 68-page Ebook from the MIT Center for Digital Business by Erik Brynjolfsson and Andrew McAfee (2011)
- It discusses the role of skills and opportunities in a digital age where computing capabilities have become significantly powerful
- It includes this quote:

  [N]ew research by David Autor and David Dorn [found] that the relationship between skills and wages has recently become U-shaped. In the most recent decade, demand has fallen most for those in the middle of the skill distribution.

What does that means to you?

- If tasks in data science seem automatable, they will
- AutoML (discussed next) is coming to get many data scientists jobs
  - The U-shaped distribution means the jobs at risk are not the ones with lowest-skills
- This is the rest of the quote:

  The highest-skilled workers have done well, but interestingly those with the lowest skills have suffered less than those with average skills, reflecting a polarization of labor demand.

FUTURE IS HYBRID

- The book puts forth that the future of employment is hybrid
- Tasks we once considered exclusive to humans are becoming more and more hybrid
- Take for example, the case of translating documents from one human language to another
  - Nowadays, more and more translation work revolves around per-word edits of machine translated drafts
  - Job satisfaction is not very high, though

You need to make your human side count

- Go for the algorithms
  - Machine Learning / Data Engineering
  - Computer Science background and very focused interests
- Go for the data (domain)
  - Improve the input
- Go for the problem (business)
  - Improve the output

Improve the input

- Do not go gentle into that good night (poem from Dylan Thomas)

  Do not go gentle into that good night,
  Old age should burn and rave at close of day;
  Rage, rage against the dying of the light.

- Likewise, rage, rage against the input data you were provided
  - Seek to better understand it
  - Find its errors
    - For example, missing data, find clever ways to fill those holes (data imputation)
  - Find more information about the processes that produced the data
  - Seek to expand the provided data with other sources of information

Expanding Features

- Poor domain modelling results in too many features
  - Feature reduction (e.g., feature selection) is key
- However, many times adding new features can have plenty of value
- Expansion includes
  - Computable features
    - That is, by computing new features using the existing ones as input
    - These small programs encode domain knowledge about the problem and dataset
  - Imputation
  - Smoothing

Expanding Features using External Data

- Many times there will be value to add external data from separate sources
- This is equivalent to add common sense knowledge
- For example, if the problem involves Geographical data, adding distance to major cities might help
  - For example, if the target is to predict fuel prediction during a road trip
- This is very domain dependent

Full example: `https://github.com/DrDub/artfeateng/blob/master/Chapter10.ipynb`, Cell #25

Handling Missing Data

- Many times the data has instances where a particular feature value is unknown
- This might be due to artifacts on the acquisition, merging from different data sources or limitations with the feature extractors
- Some machine learning libraries (e.g., Weka) have excellent functionality to deal with missing data
  - Others (e.g., scikit-learn) have very poor tools
- Beware of its impact

Imputation

- Imputation is the process of choosing what value to use to complete a missing value
- In the simpler case, the median value for the feature can be used
- In the most complex case, a classifier can be trained on the remaining features to predict the value of the missing feature
- In practice, a combination of both approaches can be used, employing a simple threshold classifier
  - But the median approach is still better than leave the feature as "0" as with scikit-learn

Full example: `https://github.com/DrDub/artfeateng/blob/master/Chapter7.ipynb`, Cell #7

Pivoting

- Another technique to decompose complex features is to apply a pivot
  - Transform data rows into an aggregated, larger tuple

Original Data

ID	Activity	Target
1	read	n
1	save	n
1	search	y
2	save	n
2	read	n
3	save	n
4	search	n
5	read	y
5	search	y

$\rightarrow$

Data after Pivot

ID	Read?	Save?	Search?	Target
1	y	y	y	y
2	y	y	n	n
3	n	y	n	n
4	n	n	y	n
5	y	n	y	y

Improve the output

- We train a model and evaluate it over held-out data or fold
- Which metrics do we use?
  - In general, the ones provided by our ML toolkits
  - It doesn't need to be that way
  - Discussing with the users of the trained model can help create a custom metric that nails the use cases for the model
- Quantitative metrics is not the only way to help obtain a more useful model
  - Do not forget about qualitative evaluation
  - Have users (and experts) see the result and discuss whether any improvements are needed

Metrics

- Metrics in general need to align with business needs
- Working with multiple metrics is hard because it is difficult to optimize multiple objective functions
  - Computers are bad at that
  - Humans do it all the time (work-life balance? politics?)
- In Question Answering, we computed 36 metrics on the results
  - Some obvious metrics such as "how many questions did we answer right?"
  - But many exotic things like "average of 1/r where r is the rank of the first correct answer in the list of answers returned for each question"

Examples of improved output

- In Question Answering, it is possible to use a system in which "I don't know" is a valid answer
- Then it is possible to prefer a system that remains quiet rather than giving the wrong answer
- Changing the output from "always give an answer irrespective of how uncertain the system is about it" to "fine tune a threshold so the system answers as many questions correctly for the ones it chooses to answer" can have a dramatic impact in both the way the model is constructed, its use and its utility

What you choose to improve is what you get

- Improvements target the metrics, not only at the computational side but also a the human side
- In colonial India, the British payed per killed cobra
  - People started breeding cobras to cash in
  - Cobra's overall population actually augments
- Similar things happen in startups
  - Sets of so called "vanity" metrics that always increase (e.g., total number of visitors)
- Spend time understand the output of your model and how it fits in the larger picture
  - It is rarely captured by a number obtained from scikit-learn

What is AutoML

- If all we need to do is choose a model and some parameters, can't a machine just do it?
- Yes, in fact, it can
- The search space is huge, but many datasets behave similarly
  - Human problems tend to be similar, that is
- Large search spaces involve tons of computational power needed
  - Not necessarily great for the environment either

AutoML in a slide

- Automating the ML process for real-world applications
- Apply a process usually involving:
  1. Data pre-processing
  2. Feature Engineering
     - 2.1 Feature Extraction
     - 2.2 Computable Features
     - 2.3 Feature Selection / Dimensionality Reduction
  3. Model (ML algorithm) selection
  4. Hyperparameter tuning
  5. Automated problem checking (target leaking, faults, misconfigurations)

Example of AutoML

- By far the simplest type of AutoML is hyper-parameter / model search
- Using scikit-learn GridSearchCV:

Example

```
parameters = [{'kernel':  ['rbf'],
      'gamma':  [1e-3, 1e-4],
      'C': [1, 10, 100, 1000]},
    {'kernel':  ['linear'],
      'C': [1, 10, 100, 1000]}]
clf = GridSearchCV( SVC(), parameters,
    scoring='recall_macro' )
```

You and AutoML

- In a recent talk, a ML professor at a major university claimed that in 10 years there will not be any "rogue" ML
- I agree that “AI” or “ML” will no longer be applied to the type of probabilistic programming resulting from heavy feature engineering over small datasets
  - But I do not expect it to disappear, just leave the AI field
  - That is common in AI
    - There was a time when “search” (as in “find a string in a document”) was part of AI
- The point is, researchers in the field are actively working towards making many entry-level tasks in Data Science obsolete

Domain Independence

- Careful data analysis can keep us away from bad assumptions and yield high-performing models without domain expertise
- Whether quality feature engineering can be achieved without deep domain expertise is a topic of debate
- In his talk "Data Agnosticism, Feature Engineering without domain expertise" Kaggle champion Nicholas Kridler argues that
  - "responsible data analysis and quick iterations produce high-performing predictive models"
  - Models will help us find features without domain expertise
- If you lack domain expertise, do not underestimate the **dataset expertise** you will gain by working with it.
  - In a doctor visit there are two experts at work:
    - the expert in medicine (the doctor) and
    - the expert on the patient's body (the patient themselves)

Domain Dependence

- Subject matter experts know which data and features are important
- Domain independent approaches can produce good models very quickly, but...
  - "If you want to go fast, go alone; but if you want to go far, go together."
    - And if you want to go really fast, ask somebody who knows the way.
- Tapping into domain expertise will allow you to get much farther than if you need to reinvent the wheel for every little quirk and problem in your target domain
- **Know the context around the data**

Example: NLP

- In Natural Langugage Processing and Information Retrieval, there are lists of function words that contain little to no semantic information content for theme assignment of full documents
  - The are called "stopwords"
  - For example "for, a, to, in, them, the, of, that, be, not, or, and, but"
  - In general, removing them improves the performance of a classifier
- Of course function words are very important for human understanding of documents and there are expressions only composed of stop words ("to be or not to be")

How to use domain expertise

- When domain expertise is available, how can it be incorporated to the ML/DM process?
- There are multiple ways, we will focus on Feature Engineering, discussed next

What is Feature Engineering

- Feature Engineering is the **process** of
  - **representing** a problem domain as to make it
  - **amenable for learning** techniques.
- This process involves
  - the **initial discovery** of features and their
  - **step-wise improvement**
- based on
  - **domain knowledge** and the
  - **observed performance** of a given ML algorithm over specific training data.

Good Features

1. **Informative**
   - the feature describes something that makes sense to a human
2. **Useful**
   - the feature is defined for as many instances as possible and
3. **Discriminant**
   - the feature divides instances into the different target classes.

Feature Engineering Cycle

1. Final evaluation split(Raw data)
   - *Raw data for feature enginering*
   - *Raw data for final test*
2. **Iterate** over ML cycle(*Raw data for feature engineering*)
   - *Featurizer*
   - *Trained model*
3. Featurize(*Featurizer*, *Raw data for final test*)
   - *Feature vectors for final test*
4. Evaluate(Trained model, Feature vectors for final test)
   - *Final evaluation*

- **Feature engineering can overfit terribly and harm your ML.** Use data in stages to avoid that.

Human-in-the-loop

- The Feature Engineering process mentioned before is a human-in-the-loop process
- It hinges on two key analysis:
  - Exploratory Data Analysis
  - Error Analysis
- We will discuss them in turn

Example: Feature Drilldown

Based on the Error Analysis results you might want to...

- Drop poorly performing features
  - Simpler models have less parameters
    - They are more informed over the same amount of data
- Expand good ones
  - Provide variations over them
  - Combine them
- Go back to your raw data and do more EDA on it to hypothesise new features

Exploratory Data Analysis

- Exploratory Data Analysis
- Analyze data sets to summarize their characteristics
  - Usually through visual means
- Help formulate hypothesis about the data
- In the ML case
  - Help pick a ML model
  - Help with initial featurization

When to do it

- Perform EDA every time you receive new raw data
- Resist the temptation to jump into model building right away
- At the very least, some general idea of the data is needed to train the model

- When new raw data is received regularly, a lighter EDA needs to be performed to check the assumptions captured from the original data still hold
  - Data drift is a common issue
  - Some of these checks can be automated

How to do it

- This is very data dependent
- EDA is usually taught in statistics courses
- It is possible to gain some insights by running a battery of common statistical tests and characterizing the distribution of the data
- For examples in the context of Data Science, the open source code available at my book's website contains multiple examples of EDA over graph, text, image and time data.

WikiCities

- WikiCities dataset
  - A resource put together for the book
  - Goal: exercising different Feature Engineering techniques
- Predicting the population of towns and cities based on:
  - Their properties
  - Their textual description
  - Their historical population
  - Their satellite imagery
- `http://artoffeatureengineering.com/code.html`

WikiCities Text

- The total text for all cities spans 43,909,804 words and over 270,902,780 characters
  - Average of 558 words per city
  - Needs aggressive feature selection
- Many Wikipedia pages contain the population information mentioned within the text. Not necessarily all of them, but many do
  - At the Exploratory Data Analysis stage we might want to get an idea of how many do
  - Even for the ones that do, however, it might be indicated in many different ways, including punctuation (2,152,111 instead of 2152111) but most probably rounded up and expressed intermixing digits with words (like *"a little over 2 million"*)
  - This task is representative of the NLP subfield of **Information Extraction**, where custom systems are built to extract specific types of information from texts with a specific style and type

EDA

- Bigger cities will have longer pages
  - text length shows encouraging results
  - 10 cities at random, most pages mention the population with punctuation
    - On to the whole dataset that percentage reduces to about half

City	Pop.	Text Highlight
Arizona City		The population was **10,475** at the 2010 census.
Hangzhou		Hangzhou prefecture had a registered population of **9,018,000** in 2015.
Zhlobin		As of 2012, the population is **80.200**.
Cournonsec	2149	-
Scorbé-Clairvaux	2412	-
Isseksi		At the time of the 2004 census, the commune had a total population of **2000** people living in 310 households.

Error Analysis

- Looking at aggregate metrics is a good start but for feature engineering you want to look in detail
- Identify individual erroneous instances or classes of instances that contribute substantively to the error
  - Using the contingency table
  - And a properly instrumented system
- Drill down on hypothesis for the reasons behind the error
- Assess whether such hypothesis will *actually* cover a lot of error cases
  - Do not overoptimize
  - Do not overengineer

Who should do it

- Error analysis is a great moment to involve the future users of the trained model
- Their feedback might go beyond errors in the model itself
  - It might be possible, for example, that the model being built solves the wrong problem altogether
  - Communication is much easier over real output from a trained model

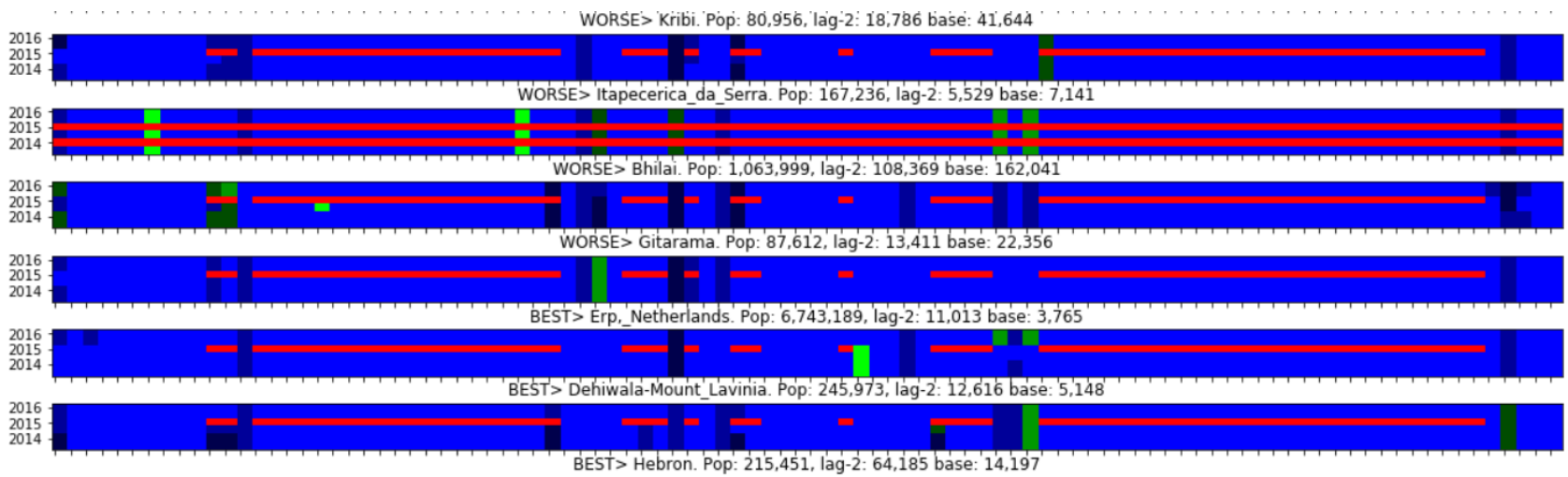
Error Analysis

`http://artoffeatureengineering.com/notebooks/Chapter8.html, # CELL 9`

City	Comments
Bailadores	A big city (over 600,000 inhabitants) described as a "town."
Villa Alvarez	No text
Koro	"Agriculture" most probably is linked to smaller places.
Curug	We got a toknumseg6 for the 2010 census and then the correct toknumseg30, I think the first toknumseg6 is confusing the ML.
Delgado	No usable terms.
Madina	Population of toknumseg30 should be gotten by a stop-words plus bigram.
Banha	I think if "cities" were "city" it will work.
Dunmore	Small hamlet with lots of info in Wikipedia, the main signal "village" is not part of the feature vector.

Example of EA (2)

http://artoffeatureengineering.com/notebooks/Chapter7.html, # CELL 12



Trading Amount of Data with Human Time

- In computer science many times, it is possible to trade **time** with **space** and viceversa
  - Speed things up by using a cache (that uses more RAM)
  - Use less RAM by doing multiple passes on the data (which takes more time)
- Similarly, if you have large amounts of data, it might be possible to dispense with Feature Engineering
  - The premise of Deep Learning
  - Synthesize better feature representations in neural network layers
- Adding human knowledge helps solving problems using computers
  - Does not help building autonomous intelligent systems

Target Rate Encoding

- If 80% of the times a binary feature is true then the binary target class is also true, instead of having the original feature as true or false, set it to 0.8
- If there are multiple target categories, have a target rate for each of the categories
- Might need to estimate these firing rates on a separate set from the training set
  - Otherwise the machine learning will trust these features too much
- An alternative is to let go of the target altogether and just represent each categorical by its frequency counts
  - The fact that it is a rare or a frequent category might be more informative than the category itself

Target Rate Encoding: Example

Dataset

F	T
A	+
B	-
C	+
A	+
B	+
C	+
A	+

→

F	T
A	+
B	-
C	-
A	+
B	-
C	-
B	+

F	TRE
A	1.0
B	**0.4**
C	*0.5*

→

TRE dataset

F_{TRE}	T
1.0	+
0.4	-
0.5	+
1.0	+
0.4	+
0.5	+
1.0	+

F_{TRE}	T
1.0	+
0.4	-
0.5	-
1.0	+
0.4	-
0.5	-
0.4	+

Full example: `https://github.com/DrDub/artfeateng/blob/master/Chapter6.ipynb`, Cell #30

STAGGING THE DATA

- At the beginning of the presentation, we discussed Xval vs. evaluating on held out
- During the discussion about overfitting, we mentioned that repeated quizzing the test data overfits multiple decisions surrounding building the ML solution to the test data
- Due to that, it is preferable to stage the release of test data
  - Given the total amount of data available, divide it into stages (e.g., 5 stages)
  - Improve the system evaluating it on stage 3 while training it on stages 1+2
  - Then stop using stage 3 for evaluation, add it to the training pool and test a model on stages 1+2+3 on stage 4
- The evaluation on a new stage should be the closest to production behaviour

ML Cycles

- In closing, we discussed
  - Basic ML cycle
  - AutoML
  - Feature Engineering cycle
    - Human-in-the-loop
  - Stagging Data

On Feature Engineering Success

- Feature engineering involves much trial and error
- In the book about 50% of the techniques discussed in the case studies were successful
  - That paints an optimistic view of feature engineering
  - A rate between 10-20% is more likely
- The same can be said from exercises in homeworks or presented in lectures
  - When you get to work on new problems and data, you will fail (a lot) even doing everything "well"

Conclusions

- In the book "Rules of Play", Zimmerman and Salen discuss that games have multiple layers
  - There are the rules for moving the chess pieces
  - Rules for playing a certain opening
  - Rules for playing with your father on a Sunday morning
- The ML/DM cycle is similar
  - At different levels, there are cycles that encompass lower-level ones
- My belief is that the ML/DL process is better done iteratively, starting from
  - Exploratory Data Analysis, and performing
  - Error Analysis, after each iteration
  - with a human-in-the-loop approach
- Tap into domain experts to go faster and achieve better results

The Art of Feature Engineering

- Focus
  - Helping practitioners
  - People already familiar with ML algos
  - Helping also approach new domains
- Material seldom presented in book format
- Coming early 2020 through Cambridge University Press
- 286 pages, 330 references, 10 chapters
- 10,000 lines of Python in 5 case studies
  - Available open source at `http://artfeateng.ca`
- Keep in touch!
  - @pablodubouе on TW