PUT DOWN THE DEEP LEARNING

When not to use neural networks (and what to do instead)

Dr. Rachael Tatman
Data Scientist Advocate @ Kaggle

GAMING ENTERTAINMENT TECH

Feeble humans prove no match for OpenAI's Dota 2 gods

The AI won 7,215 matches against humans, losing only 42 in the process

By Vlad Savov | @vladsavov | Apr 23, 2019, 9:25am EDT

MICROSOFT TECH ARTIFICIAL INTELLIGENCE

Microsoft reaches 'human parity' with new speech recognition system

By Sam Byford | @345triangle | Oct 18, 2016, 9:47pm EDT

SHARE

SCI-TECH

Google's AlphaGo Zero destroys humans all on its own

The new artificial neural network taught itself to master the ancient game Go within weeks, without any tips from humans.

BY ERIC MACK | OCTOBER 20, 2017 8:16 AM PDT

3

Google's New Robot Is Better at Tossing Things Than You Are

By Ryan Whitwam on March 27, 2019 at 2:51 pm | 7 Comments

@rctatman

Additionally, for BERT$_{\text{LARGE}}$ we found that fine-tuning was sometimes unstable on small data sets (i.e., some runs would produce degenerate results), so we ran several random restarts and selected the model that performed best on the Dev set.

(Devlin et al 2019)

GPT-2 model from OpenAI

Mario Klingemann @quasimondo

Follow

How much would it cost to train your own #BigGAN from scratch if we had the training code?

The 512x512 model requires 512 TPU v3 which cost US$ 2.40 per TPU per hour. According to the paper it takes between 24 and 48 hours to train a model.

For $59,000 it can be yours. 😱

6:17 AM - 22 Nov 2018

55 Retweets 253 Likes

15 55 253

stormtrooper1729 @stormtroper1721 · Feb 14

Replying to @Smerity @Google @OpenAI

How long do you think they ran the models for? A week? That would come out to almost $15k

1

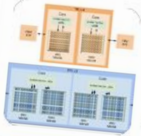
Smerity @Smerity · Feb 14

If it was a week it'd be $8 x 256 x (7 x 24) = $344,064. Without any additional information it's hard to even guess how long training might take. They also report in the paper that it's still underfitting so they'd likely happily let it sit there for as long as they could.

2 1

Smerity @Smerity · Feb 14

Gah, sorry, to clarify, it's 8 times less ($43,008) as I mistook cores for TPUs. Still a stupendous amount!



Smerity @Smerity

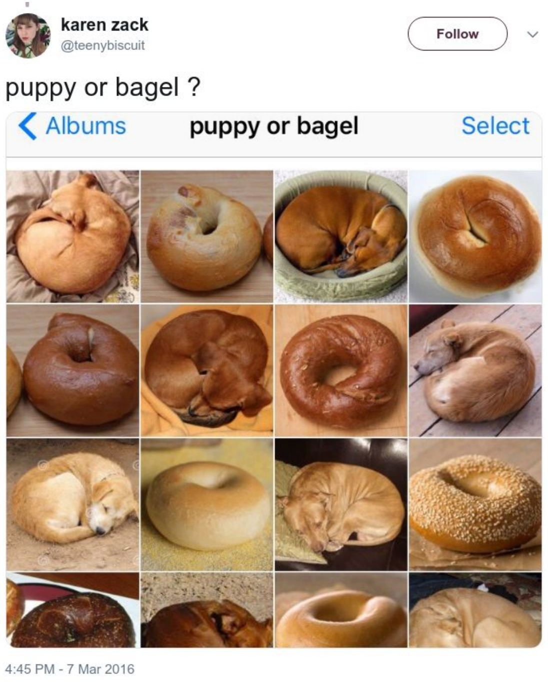
Gah, you're right, it's 256 cores and not 256 TPUs. After double checking (TPUv2 has 2 cores per chip vs TPUv3 with 4 cores), I'm off by 8x and Twitter as ever won't allow edits -_-...

1 2

@rctatman

I would personally use deep learning if...

- A human can do the same task extremely quickly (<1 second)
- I have high tolerance for weird errors
- I don't need to explain myself
- I have a large quantity of labelled data (>5,000 items per class)
- I've got a lot of time (for training) and money (for annotation and compute)



@rctatman

Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot

@rctatman

Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot
Regression			
Trees			
Distance Based			

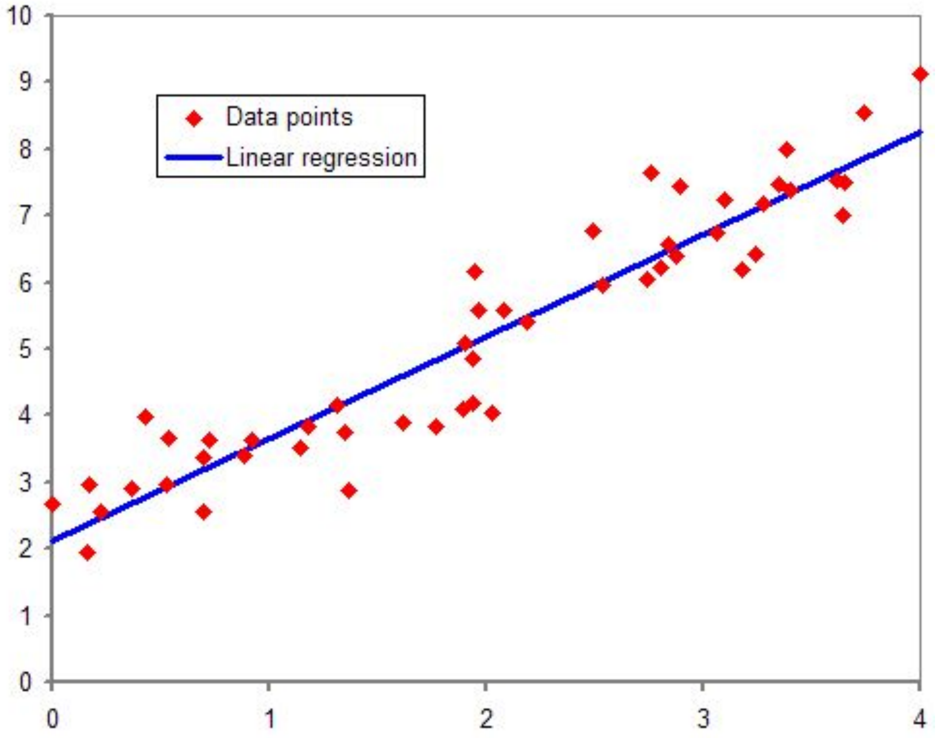
@rctatman

Regression 

The OG ML technique

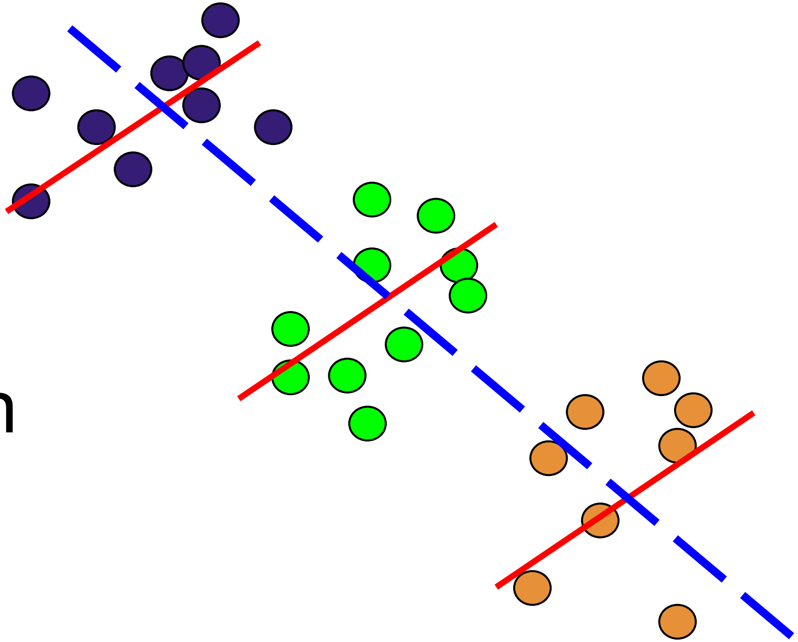
- In regression, **you pick the family of the function** you'll use to model your data
- Many existing kinds of regression models

✓ Fast to fit
✓ Works well with small data
✓ Easy to interpret
✘ More data preparation
✘ Models require validation



@rctatman

My go-to?
Mixed effects regression



@rctatman

```python
# imports for mixed effect libraries
import statsmodels.api as sm
import statsmodels.formula.api as smf

# model that predicts chance of admission based on
# GRE & TOEFL score,with university rating as a random effect
md = smf.mixedlm("chance_of_admit ~ gre_score + toefl_score",
                 train, # training data
                  groups=train["university_rating"])

# fit model
fitted_model = md.fit()
```

@rctatman

```
Mixed Linear Model Regression Results
=============================================================
Model:               MixedLM Dependent Variable: chance_of_admit
No. Observations: 300     Method:                 REML
No. Groups:          5       Scale:                  0.0055
Min. group size:  21      Likelihood:             332.7188
Max. group size:  99      Converged:              Yes
Mean group size:  60.0
--------------------------------------------------------------
              Coef.  Std.Err.    z    P>|z| [0.025  0.975]
--------------------------------------------------------------
Intercept    -1.703    0.169 -10.097 0.000 -2.033  -1.372
gre_score     0.005    0.001   7.797 0.000  0.004   0.007
toefl_score   0.007    0.001   4.810 0.000  0.004   0.009
Group Var     0.002    0.020
```

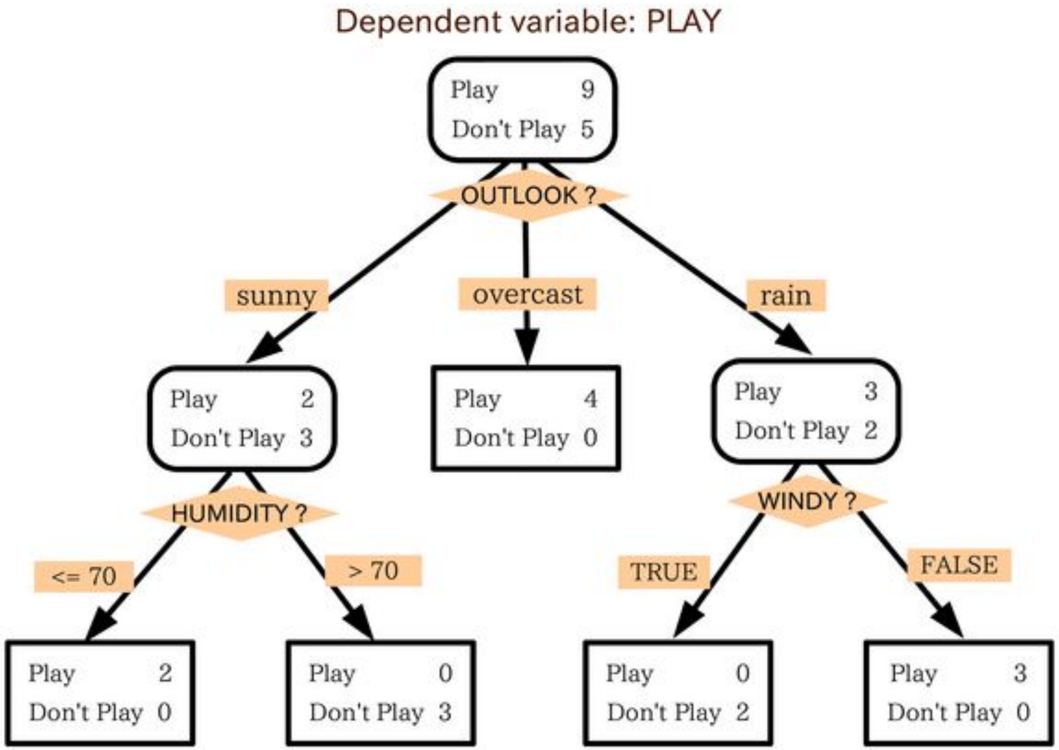
@rctatman

Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot
Regression	Some	A little	A little
Trees			
Distance Based			

@rctatman

Trees 🌳

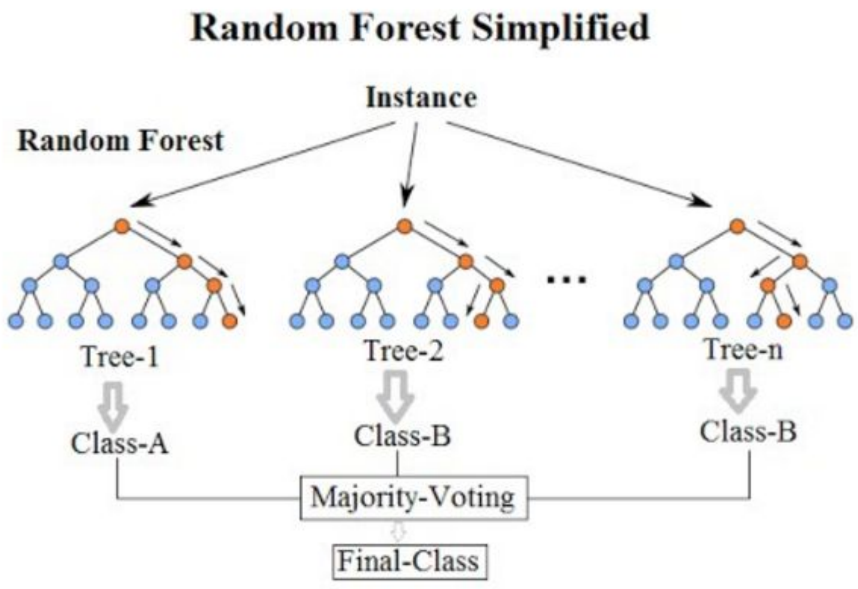
Tree based methods



@rctatman

Random Forests

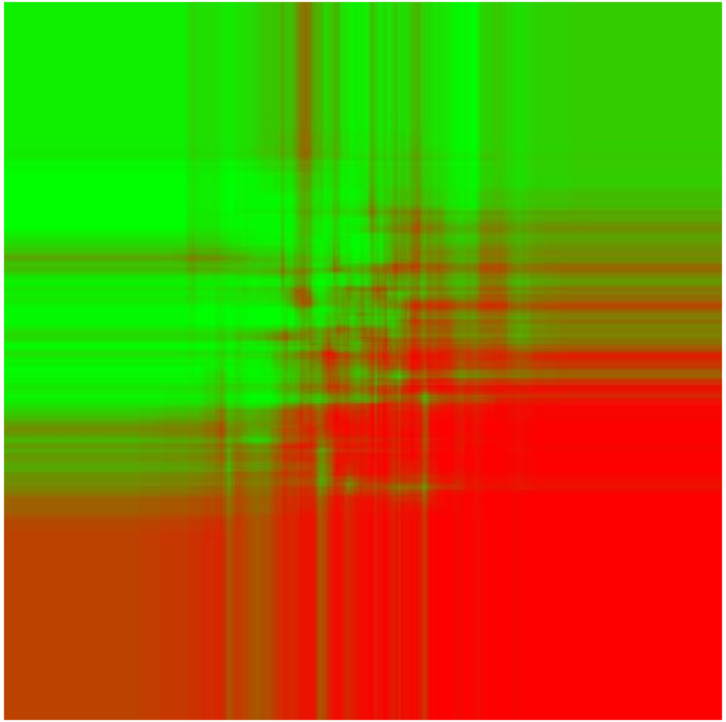
- An **ensemble** model that combines many trees into a single model
- Very popular, especially with Kaggle competitors
  - **63% of Kaggle Winners (2010-2016) used random forests,** only 43% deep learning
- Tend to have better performance than logistic regression
  - "[Random forest versus logistic regression: a large-scale benchmark experiment]", Couronné et al 2018



@rctatman

Benefits & Drawbacks

- ✓ Require less data cleaning & model validation
- ✓ Many easy to use packages
    - XGBoost, LightGBM, CatBoost, new one in next scikit-learn release candidate
- ❌ Can overfit
- ❌ Generally more sensitive to differences between datasets
- ❌ Less interpretable than regression
- ❌ Especially for ensembles, can require more compute/training time



@rctatman

```python
import xgboost as xgb

# split training data into inputs & outputs
X = train.drop(["chance_of_admit"], axis=1)
Y = train["chance_of_admit"]

# specify model (xgboost defaults are generally fine)
model = xgb.XGBRegressor()

# fit our model
model.fit(y=Y, X=X)
```

@rctatman

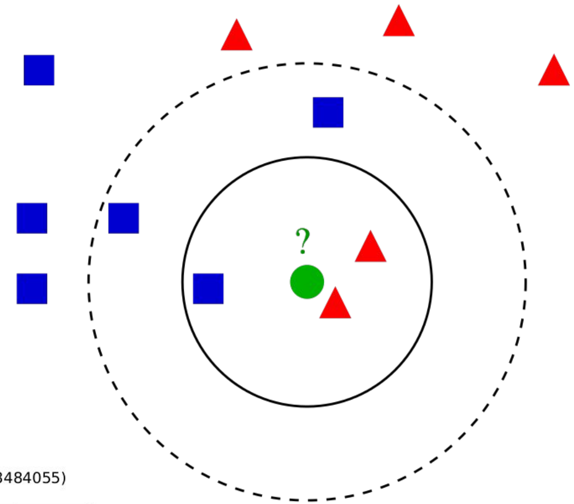
Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot
Regression	Some	A little	A little
Trees	Some (esp for big ensembles)	A little	Some
Distance Based			

@rctatman

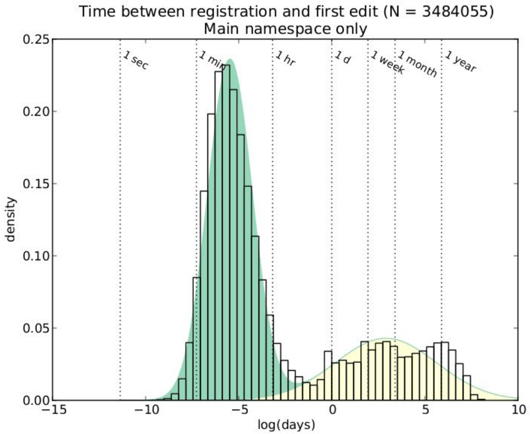
Distance 🔢

Distance based methods

- **Basic idea:** points closer together to each other in feature space are more likely to be in the same group
- Some examples:
  - K-nearest neighbors
  - Gaussian Mixture Models
  - Support Vector Machines



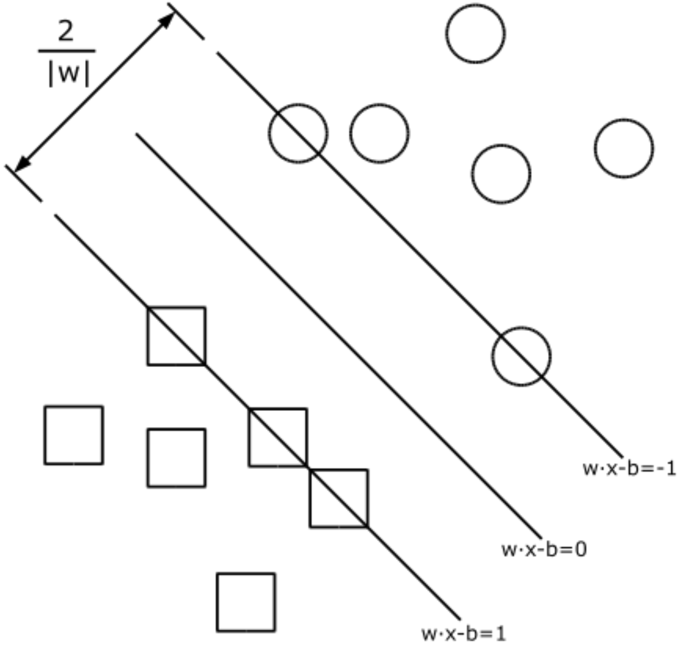
Antti Ajanki AnAj [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/)]



Junkie.dolphin [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)]

Benefits & Drawbacks

- ✓ Work well with small datasets
- ✓ Tend to be *very* fast to train
- ❌ Overall accuracy is fine, other methods usually better
- ❌ Good at classification, generally crummy/slow at estimation
- These days, tend to show up mostly in ensembles
- Can be a good fast first pass at a problem



@rctatman

```python
from sklearn.svm import SVR

# split training data into inputs & outputs
X = train.drop(["chance_of_admit"], axis=1)
Y = train["chance_of_admit"]

# specify hyperparameters for regression model
model = SVR(gamma='scale', C=1.0, epsilon=0.2)

# fit our model
model.fit(y=Y, X=X)
```

@rctatman

Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot
Regression	Some	A little	A little
Trees	Some (esp for big ensembles)	A little	Some
Distance Based	Very little	Very little	Very little

So what method should you use?

Method	Time	Money	Data
Deep Learning	A lot	A lot	A lot
Regression	Some	A little	A little
Trees	Some (esp for big ensembles)	A little	Some
Distance Based	Very little	Very little	Very little

Method	Time	Money	Data	Performance (Ideal case)
Deep Learning	A lot	A lot	A lot	Very high
Regression	Some	A little	A little	Medium
Trees	Some	A little	Some	High
Distance Based	Very little	Very little	Very little	So-so

Method	Time	Money	Data	Performance (Ideal case)
Deep Learning	[illegible]	[illegible]	A lot	Very high
Regression	[illegible]	[illegible]	little	Medium
Trees	[illegible]	[illegible]	Some	High
Distance Based	[illegible]	[illegible]	Very little	So-so

Most Powerful

Most Interpretable

User Friendliest

Most Lightweight

Data Science != Deep Learning

- Deep learning is extremely powerful but it's not for everything
- Don't be a person with a hammer
- Deep learning isn't the core skill in professional data science
  - "I always find it interesting how little demand there is for DL skills... Out of >400 postings so far, there are 5 containing either PyTorch, TensorFlow, Deep Learning or Keras" -- Dan Becker

Thanks!
Questions?

Code & Slides:

https://www.kaggle.com/rtatman/non-deep-learning-approaches

http://www.rctatman.com/talks/

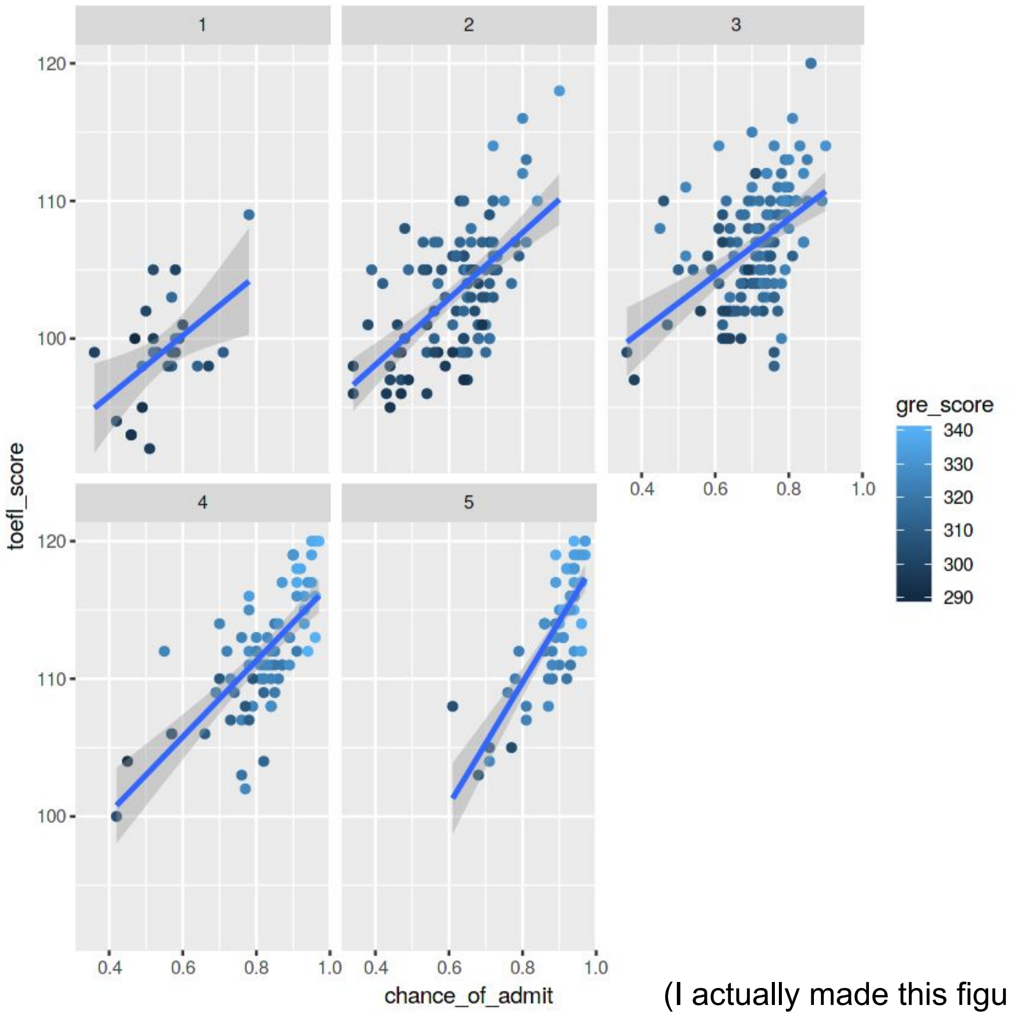
Honorable mention:
Plain ol’ rules

Sometimes ✋ Hand-Built ✋ Rules are Best

Some examples of proposed deep learning projects from the Kaggle forums that should probably be rule-based systems:

- Convert Roman numerals (IX, VII) to Hindu-Arabic numerals (9, 7)
- Automate clicking the same three buttons in a GUI in the same order
- Given a graph, figure out if a list of nodes is a valid path through it
- Correctly parse dates from text (e.g. “tomorrow”, “today”)

Remember: If it’s stupid but it works, it’s not stupid.



(I actually made this figure in R 🤫)

@rctatman