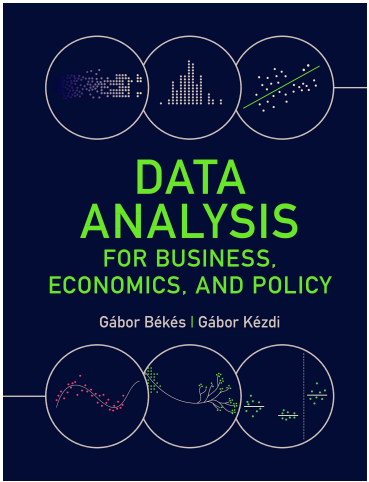
2. Prediction process

Gabor Bekes

Data Analysis 3: Prediction

2021

Slideshow for the Békés-Kézdi Data Analysis textbook

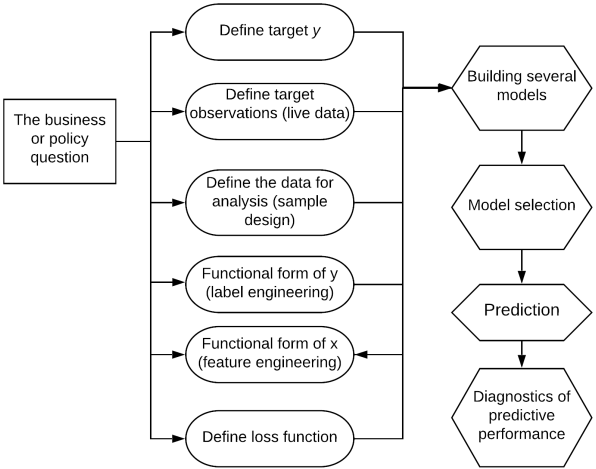


- Cambridge University Press, 2021 April
- Available in paperback, hardcover and e-book
- **gabors-data-analysis.com**
  - Download all data and code `https://gabors-data-analysis.com/data-and-code/`
- This slideshow is for **Chapter 14**
  - Slideshow be used and modified for educational purposes only

1. Business question and defining y

- The first task is defining the business or policy question we seek to answer.
  - What kind of car or hotel prices are we interested in?
  - How is our decision related to monetary or other rewards?
- Answers will guide any further action.
- Design process of the analysis

Steps of Prediction



Sample design

- In a prediction exercise, we are interested in predicting target for a set of units we care about and are less involved in generalization.
- The fact that we are less interested in inference and more in prediction makes working on our sample a bit more important.

Sample design: filtering

- Before settling on a model, we need to design the sample.
- Filtering our data to match the business/ policy question.
- It may involve dropping observations based on key predictor values,
  - If may not be interested in predicting values for all the cars just personal vehicles.
  - We would be looking for 3-4 star hotels, not all of them.

Sample design: Spotting errors

- For prediction exercise, we should spend more time on finding and deleting errors.
- We have no chance predicting extreme values, and certainly not errors.
- Actual parameter estimates will be important.
- Hence, keeping an extreme value that is likely to be an error, will have a high cost - the quadratic errors in the loss function will tilt the curve and our prediction will be off for most observations.
- Stronger focus on dropping observations we think are errors.

Case study of used cars: Sample design

- dropping hybrid cars, manual gear, truck
- drop cars without a clean title (i.e., cars that had to be removed from registration due to a major accident)
- drop when suspect cars with clearly erroneous data on miles run,
- drop cars in a fair (=bad) condition, cars that are new
- Data cleaning resulted in 281 observations

Label engineering - defining target

- We need to define what will be our target variable.
- In some cases, this requires no action,
  - The business question may define it: the price of the hotel is one such case.
- Often it requires thinking and decision-making about definition.
  - How to define default, injury, purchase
- Binary vs continuous.
  - If interested in classifying firms by survival and default, we need to define the time horizon, and the label for zombie firms - not in default, but not in operation either. (Week 3)
- Log vs level

Label engineering - log vs level

- When price is the target variable, its relation to predictor variables is often closer to linear when expressed in log price.
- Both substantive and technical reasons.
  - Log differences approximate relative, or percentage, differences, and relative price differences are often more stable.
  - The related technical advantage is that the distribution of log prices is often close to normal, which makes linear regressions give better approximation to average differences.
- Keeping our target in level or transforming into a log value is an important modelling choice.
  - Not straightforward.

Label engineering - log vs level

- Importantly, when the target variable is expressed in log terms, we want to predict the value of the target variable ($\hat{y}$) not its log ($\widehat{\ln y}$).
- One may simply raise e to the power of the predicted log target variable : $\hat{y} = e^{\widehat{\ln y}}$.
- Not enough! One has to adjust this power by a function of the standard deviation of the regression residual $\hat{\sigma}$

$$\hat{y}_j = e^{\widehat{\ln y}_j} e^{\hat{\sigma}^2/2} \tag{1}$$

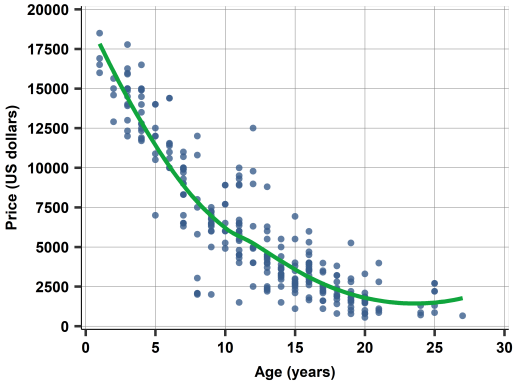
- Why correction term?
  - Regression predicts average (expected) ln *price*
  - We need average (expected) *price*
  - But average *exp*(*lnprice*) is not the same as average *price*
  - The ln function is concave

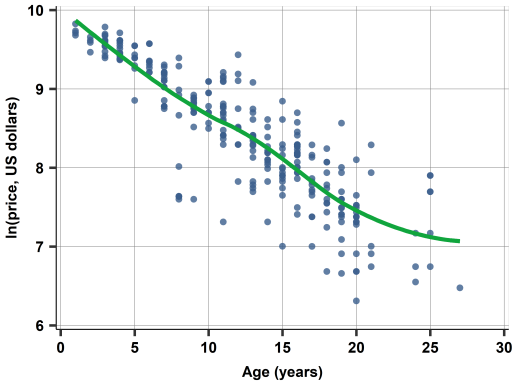
Used cars case study: Label engineering - log?

- Business case is about price itself, continuous
- But model can have level or log price as target
  - Look at some patterns
  - Compare model performance
- Log vs level model - some coefficients easier interpreted
  - When we have two cars of same age and type; the one with 10% more miles in the odometer is predicted to be sold for 0.5% less.
  - SE version is 1300 dollar more costly.
- In this case, level is okay.

Case study: Label engineering - log?

Level with quadratic or linear in logs - both options of modelling seems okay.





Feature engineering

- Requires the most effort
- Feature engineering - defining the list and functional form of variables we will consider as predictor.
- Importantly, we use both domain knowledge - information about the actual market, product or the society - and statistics to make decisions.

Feature engineering - checklist

This is the most essential and impacting part of prediction with regressions. Here is a quick checklist of tasks

1. What to do with missing values
2. Dealing with ordered categorical values - continuous or set of binaries
3. How to use text to create variables
4. Selecting functional form
5. Thinking interactions

What to do with missing values (recap from DA1)

- Missing at random
  - Observations with missing variables are not systematically different from rest
  - Missing values are from same distribution as observed one
- If random, may replace it with sample mean – to avoid lost observations. Add binary flag variable (x_flag =1 if imputed 0 otherwise), add it to model
  - If categorical, add N/A as a new value
- Missing systematically, by nonrandom selection
  - Must analyze reasons
  - Missing may simply mean =0
  - Look at the source of the data / questionnaire
- If very few missing and it is random
  - do not do anything. Your of obs will drop a bit, but that's it

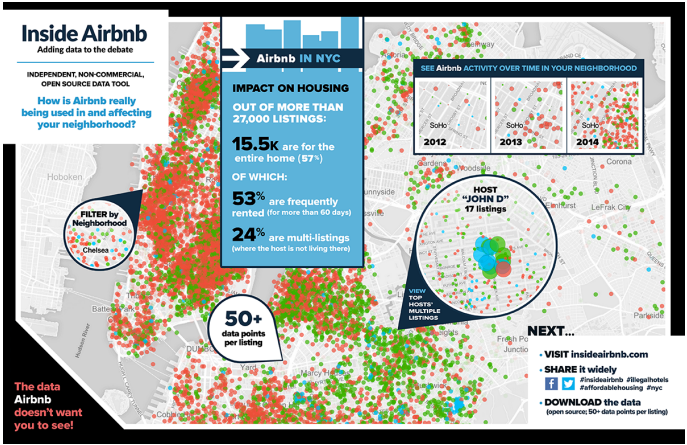
What to do with different type of variables

- Type of variable
- If binary (e.g, yes/no; male/female; 1/2) – create a 0/1 binary variable
- If string / factor – check values, and create a set of binaries.
  - Often: key task will be to merge outcomes for the purpose of parsimony. Science + art
- Continuous – nothing to do. Make sure it is stored as number
- Text – Natural Language Processing. Mining the text to get useful info.
  - May simply go and find some words, and create binaries –> seminar
  - Everything else is complicated

Case study: Predicting Airbnb Apartment Prices - Intro

- Predicting Airbnb Apartment Prices: Selecting a Regression Model
- The goal is to predict the price that may be appropriate for an apartment with certain features.
- Business case: have apartments, need to price them for rent

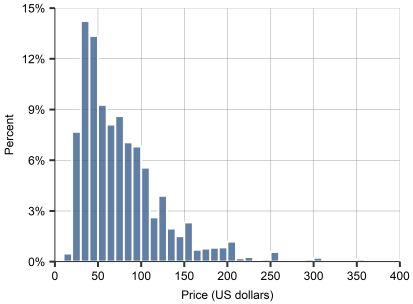
Case study: Airbnb



Airbnb prices

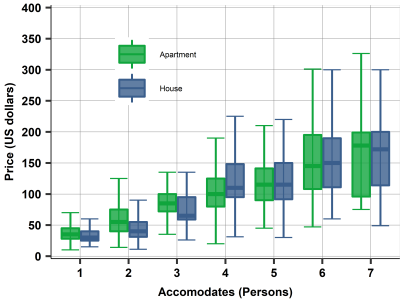
- London,UK
- `http://insideairbnb.com`
- 50K observations
- 94 variables, including many binaries for location and amenities
- Key variables: size, type, location, amenities
- Quantitative target: - price (in USD)

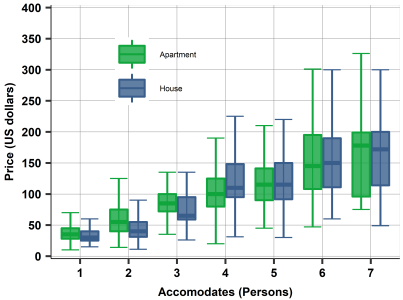
Airbnb prices



- Today: focus on a borough of Hackney
- N= 5K observations
- Price distribution (<400 USD)
  - Today: do it in level, could do logs
- Dropped very large apartments. Our final original data has 4393 observations.

Airbnb apartment price distribution by important features

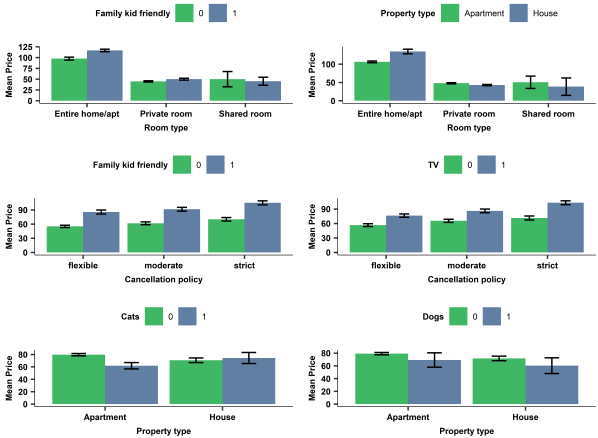




Case Study: Airbnb Feature engineering

- Key issue is to look at variables and think functional form
- Guests to accommodate goes up to 16, but most apartments accommodate 1 through 7. Keep as is. Add variables for type. No need for complicated models
- Regarding other predictors, we have several binary variables, which we kept as they were: type of bed, type of property (apartment, house, room), cancellation policy.
- Look at possible need for interactions by domain knowledge / visualization

Case Study: Graphical way of finding interactions



Model building

- Model building is essentially deciding about the predictors to include in the model and their functional form.
- We have strong computers, cloud, etc - why could not we try out all possible models and pick the best one?

We Can't Try Out All Possible Models

- We have N observations and p predictors
- Main reason for model selection problem is that we can not try out every potential combination of models.
- As p increase, trying out all options becomes prohibitively complicated and computationally intractable.
- This types of problems in computer science are called **NP-hard**
- NP stands for "non-deterministic polynomial acceptable" problems.
- The consequence of this is rather important: There is no silver bullet in feature selection.

Model building

Two methods to build models:

- by hand - mix domain knowledge and statistics
- by smart algorithms = machine learning

Model building and selection: Build model by hand

- Use domain knowledge drives picking key variables
- Drop garbage - comb through your variables and drop those that are useless. May be because of poor coverage or quality, or they may be irrelevant.
- Look at a pairwise correlations. Multi-collinearity is an issue for smaller datasets
- Prefer variables that are easier to update - cheaper operation of a prediction model used in production
- What you do will be driven by sample size and relative number of variables.
- Matters when you have relatively many variables compared to size of observations
  - Such as 100 variables, 3000 observations.

Selecting Variables in Regressions by LASSO

- Key question: which features to enter into model, how to select?
- By hand – domain knowledge. Advantage: interpretation, external validity
  - Disadvantage: with many features it's very hard. Esp. with many possible interactions

- There is room for an automatic selection process.
- Some are computationally very intensive (compare every option?)
- Advantage: no need to use outside info
- Disadvantage: may be sensitive to overfitting, hard to interpret

LASSO

- *LASSO* (the acronym of Least Absolute Shrinkage and Selection Operator) is a method to *select variables to include in a linear regression* to produce good predictions and avoid overfitting.
- It starts with a large set of potential predictor variables that, typically, include many interactions, polynomials for nonlinear patterns, etc.
- LASSO modifies the way regression coefficients are estimated by adding a penalty term for too many coefficients.
- The way its penalty works makes LASSO assign zero coefficients to variables whose inclusion does not improve the fit of the regression much.
- Assigning zero coefficients to some variables means not including them in the regression.

LASSO

- The spirit is similar to the adjusted in-sample measures of fit, such as the BIC.
- LASSO finds a regression that balances fitting the data and the number of variables.
- But its result is different: instead of producing a better measure of fit for any regression model to help find the best one it produces a better regression directly.

LASSO

Consider the linear regression with i=1...n observations and k variables, denoted 1...k:

$$y^E = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k = \beta_0 + \sum_{j=1}^{k} \beta_j x_j \tag{2}$$

Coefficients are estimated by OLS: which minimizes the sum of squared residuals:

$$\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - (\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}))^2 \right\} \tag{3}$$

LASSO modifies this minimization by a penalty term:

$$\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - (\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}))^2 + \lambda \sum_{j=1}^{k} |\beta_j| \right\} \tag{4}$$

LASSO: how it works

- λ in the formula above is called the tuning parameter.
- It serves as a weight for the penalty term versus the OLS fit. Thus, it drives the strength of the variable selection
- Perhaps surprisingly, the main effect of this constraint is to force many coefficients to zero.
- Intuitively, that is because the best way to keep the sum of the absolute value of the coefficients low while maximizing fit is to put zero values for the coefficients on many variables whose inclusion improves fit only a little.
- This adjustment gets rid of the weakest predictors.

LASSO: how it works

- The value of the tuning parameter λ drives the strength of this selection.
- Larger λ values lead to more aggressive selection and thus fewer variables left in the regression.
- But how can one specify a λ value that leads to the best prediction?
- We don't need, the algorithm does
- The LASSO algorithm can numerically solve for coefficients and the λ parameter at once.
  - This makes it fast.
- Unlike OLS, we have no closed form solutions.

Case Study: Airbnb Pricing Model building

- Process: build many models that differ in terms of features:
  - Which predictors are included
  - Functional form of predictors
- Here: specified eight linear regression models for predicting price.

- Data has 4393 observations. This is our original data.
  - 80% is our work set (3515 observations), the rest we will use for diagnostics.
  - Work set will be used for cross-validation with several folds of training and test sets.

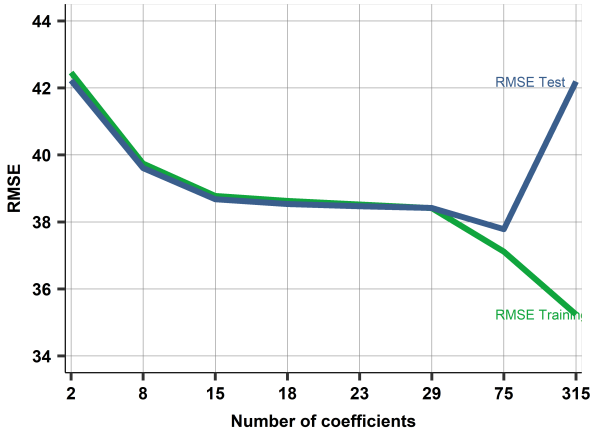
Case Study: Airbnb price prediction models

Mod	Predictor variables	N var	N coeff
M1	guests accommodated, linearly	1	2
M2	= M1 + N beds, N days review, type: property, room, bed type	6	8
M3	= M2 + bathroom, cancellation, review score, N reviews (3 cat)+ F(miss)	11	16
M4	= M3 + N guest squared, square+cubic for days since 1st review	11	17
M5	= M4 + room type + N reviews interacted with property type	11	25
M6	=M5 + air conditioning, pets allowed - interacted with property type	13	30
M7	=M6 + all other amenities	70	87
M8	=M7 + all other amenities interacted with property type + bed type	70	315

Case Study: Airbnb apartment price prediction models

Model	Coefficients	R-squared	BIC	Training RMSE	Test RMSE
M1	2	0.38	36356	42.46	42.23
M2	8	0.46	35941	39.74	39.61
M3	16	0.48	35827	38.78	38.68
M4	19	0.49	35824	38.63	38.53
M5	25	0.49	35846	38.52	38.47
M6	30	0.49	35877	38.41	38.42
M7	87	0.53	36025	37.12	37.78
M8	315	0.56	37679	35.24	42.19

Case Study: Training and test set RMSE for eight models



Case Study: Training and test set RMSE for eight models

- Training RMSE falls with complexity
- Test RMSE falls then rises

- We pick Model M7 based on lowest CV RMSE..

QUIZ

Case Study: The LASSO model

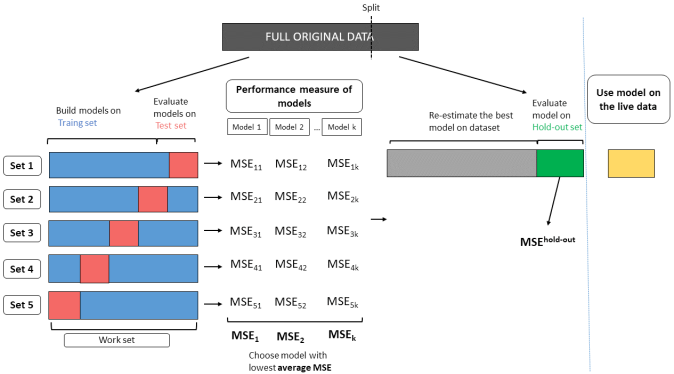
- Start with M8 and 300+ candidate variables in the regression.
- We ran the LASSO algorithm with 5-fold cross-validation for selecting the optimal value for λ.
- The algorithm picked a regression with 81 variables. Close to our model M7
- RMSE for the LASSO regression is 37.60. It is 37.78 for M7.
- Here: LASSO regression is not better but: LASSO is automatic, a great advantage.
- Here: domain knowledge helped create M7. In other cases, LASSO could be great.

Evaluating the Prediction Using a Holdout Set

- Model selection: selecting the best model using cross-validation
- Once we have picked the best model, we advised going back and using the entire original data for the final estimate and to make a prediction.
- What part of the data should we use to evaluate that final prediction?
- we are interested in how good that model would be at predicting y in the live data.
- Assume that the live data is similar to what we have.

- The solution is a random split before we do the analysis.
- Work set: We do all of the work using one part of the data: model building, selecting the best model and then making the prediction itself.
- Holdout set: another part of the data for evaluating the prediction itself. Don't touch till the end.

Illustration of the uses of the original data and the live data



The holdout set

- To do diagnostics and give a good estimate of how the model may work in the live data
- Additional twist to the process
- The holdout set.
- Holdout set is set is not used in any way for modelling – taken out in the beginning
- Used to give best guess for performance in live data
- Used to do diagnostics of our model

The holdout set

- Actual dataset
- Work set + holdout set
- Work is for 5-fold CV with training set 1,2... 5 and test set 1,2,...5.

- Live data is the data we'll use our model but do not have.

- Other names used (such as validation for test, test for holdout).

Post-prediction diagnostics

- Post-prediction diagnostics - understand better how our model works
- We look at prediction interval to learn about what precision we may expect to see of the estimates.
- We look at how the model work for different classes of observations
  - such as young and old cars.

Cross-validation and holdout set procedure

Using a subsample of the original data for various steps.

1. Starting with the original data, split it into a larger work set and a smaller holdout set.
2. Further split the work set into training sets and test sets for k-fold cross-validation.
3. Build models and select the best model using that training-test split.
4. Re-estimate the best model using all observations in the work set.
5. Take the estimated best model and apply it to the holdout set.
6. Evaluate the prediction using the holdout set.

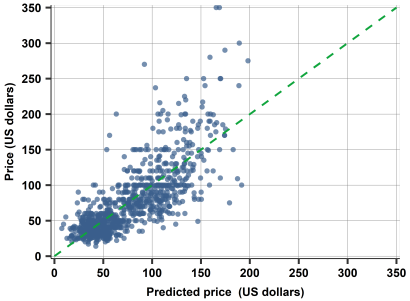
Case study: Data work and holdout

- Data has 4393 observations. This is our original data.
  - random 20% holdout set with 878 observations.
  - The remaining 80% is our work set (3515 observations).
  - Work set will be used for cross-validation with several folds of training and test sets.

Diagnostics

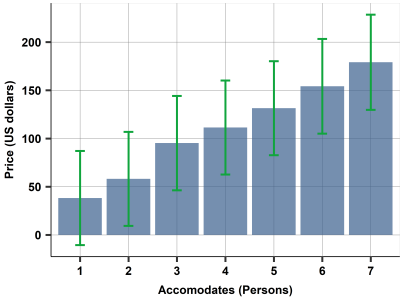
- Chose the OLS estimated M7.
- What can we say about model performance?
- After estimating the model on all observations in the work sample, we calculated its RMSE in the holdout sample. The RMSE for M7 is 31.9.
  - Smaller than CV RMSE. Luck. + Small sample size is an issue here. Why?
- Look at diagnostics on the holdout set.

Diagnostics: prices



- y-y-hat plot
- on average we are doing well (as should be). (BTW: Why?)
- higher values not really caught.

Diagnostics: variation by size



- The model generates a very wide 80% PI for average apartment - predict 77 dollar price. We face a great deal of uncertainty despite having a good model; prices may vary between 28.5 and 126.5 dollars
- bar plot with PI bands
- wide intervals
- linear and thus, hurts small number more

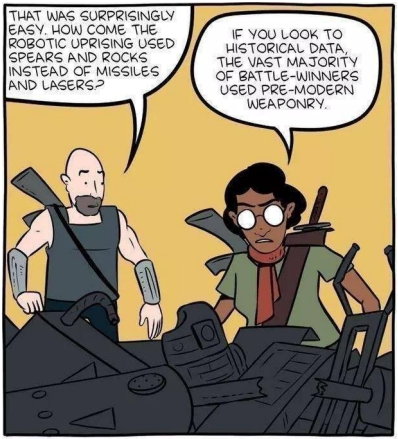
Airbnb case study: Summary

- Our aim was to build a prediction model for pricing apartments
- We built a model, M7, with domain knowledge, and a horse race between models of various complexity
  - Picked the winner by cross-validated RMSE
- The model is useful for predication, but there is a great deal of uncertainty as suggested by diagnostics (on the holdout set)

Think external validity

- Future dataset will look different
- Think about how much
- Really matters in prediction
- If uncertain, pick simpler model

External validity matters in prediction



Thanks to machine-learning algorithms, the robot apocalypse was short-lived.