NULOGY

take home assignment

how much does the shoe cost?

written by Max Pechyonkin

Take-Home Assignment

Size of data

19315 rows × 43 columns

✓ ①. read in data ← had corrupted rows

✓ ②. explore the data

✓ ③. discard fully NaNs

✓ ④. extract features from parsable columns

this will create more colums (parse date strings into categorical vars)

simplify some cols (e.g. reviews → "yes" or "no")

? ⑤. remove irrelevant products (like glasses) ← don't know how to do yet

✓ ⑥. parse data, prepare for processing/encoding

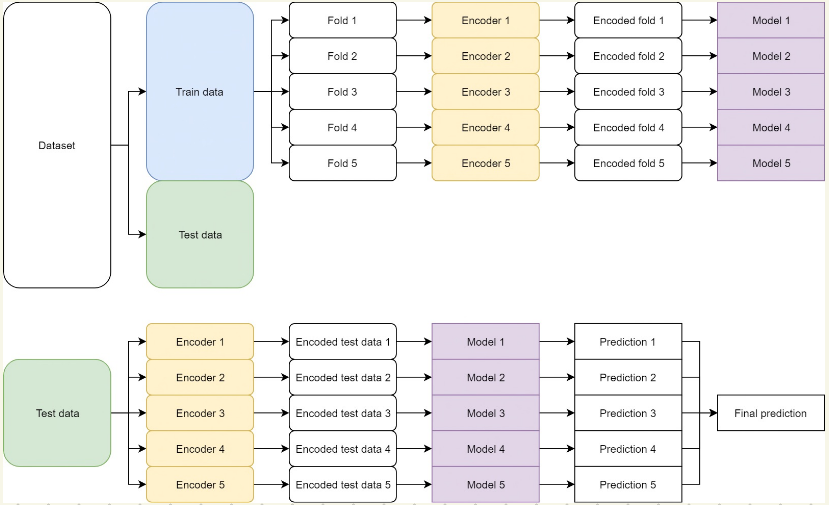
✓ ⑦. use CatBoost encoder for categorical features

✓ ⑧. Train model on all data [test]

✓ ⑨. use single validation training schedule with boosted trees

category_encoders
lightgbm
} pip dependencies

Single Validation Training



Dataset

Train data

Test data

Fold 1, Fold 2, Fold 3, Fold 4, Fold 5

Encoder 1, Encoder 2, Encoder 3, Encoder 4, Encoder 5

Encoded fold 1, Encoded fold 2, Encoded fold 3, Encoded fold 4, Encoded fold 5

Model 1, Model 2, Model 3, Model 4, Model 5

Test data

Encoder 1, Encoder 2, Encoder 3, Encoder 4, Encoder 5

Encoded test data 1, Encoded test data 2, Encoded test data 3, Encoded test data 4, Encoded test data 5

Model 1, Model 2, Model 3, Model 4, Model 5

Prediction 1, Prediction 2, Prediction 3, Prediction 4, Prediction 5

Final prediction

Feature Analysis

Legend

- 🔴 remove feature
- 🟣 convert to numerical variable
- 🔵 convert to categorical variable
- 🟢 investigate / additional work required
- 🟠 error in parsing by Pandas

feature	%NaNs	n_unique	comment
quantities prices.flavor prices.source prices.count prices.warranty prices.availability	> 99.3%	doesn't matter	get rid of this variable ∅
✓ prices.size 🔵	96.6%	64	str → cat
✓ prices.color 🔵	96.15%	66	str → cat
✓ weight 🔴🟢	95.57%	111	str → a mess of units, remove?
✓ prices.returnPolicy 🔵	94.91%	10	str → cat
✓ reviews 🟢🟣	91.41%	442	json → can parse rating ? possibly NLP but not within time constraint [investigate] [numericalize]
✓ asins 🔴	86.68%	885	csv string, Amazon identifier → remove (don't want to fit on ID)
✓ dimension 🟢	84.51%	277	str → (fairly well structured) → parse [investigate]
✓ prices.shipping 🔵🟢	70.35%	316	str → messy mess → categorize, parse [investigate]
✓ sizes 🟢🟣	69.1%	1087	csv string, messy → split, remove letters, [numericalize]
✓ prices.offer 🔵	68.77%	1244	str, messy → categorize
✓ manufacturer 🔵	65.5%	669	str → cat
✓ skus 🔴	54.76%	4450	json of SKUs (unique id) remove (don't want to train on ID)
✓ descriptions 🔴	49.05%	5118	json with values, could use NLP if more time, but remove for now
✓ ean 🔴🟠	48.41%	5653	float !incorrectly parsed from csv, must be str! remove (don't fit IDs)
✓ upc 🔴🟠	44.31%	6082	float !incorrectly parsed from csv, must be str! remove (don't fit IDs)
✓ colors 🔵🟢	42.98%	2235	csv with colors, pretty clean [investigate TfidVectorizer] Simple hack → convert to number of colors available
✓ prices.condition 🔵	34.70%	11	str → cat
✓ prices.merchant 🔵	28.66%	747	str → cat
✓ features 🔴🟢	27.90%	6539	json, messy, unstructured, needs work [investigate] remove (for now)
✓ merchants 🔵🟣🟢🔴	27.68%	6132	json, unstructured [investigate] → # of merchants, merch. name? remove for now
✓ manufacturerNumber 🔴	21.88%	6925	string remove id overfitting
✓ imageURLs 🔴	5.40%	9205	string → URL → image remove (however, I could build a DL model as predictor/feature extractor)
✓ brand 🔵	1.33%	1953	str → cat

use pandas.Series.str.get_dummies

✓	sourceURLs	●	0.10%	9956	str → URL → webpage
✓	prices.dateSeen	● ●	0.10%	1237	str → datetime → features? [investigate] remove (for now)
✓	prices.sourceURLs	●	<0.1%	11827	str → URL → webpage (missing some pages)
✓	prices.currency	● ●		10	str → currency → FX rate → common price?
✓	prices.isSale	● ●		10	true, True, false, False → mess!
✓	prices.amountMin	● ●		6651	str → float remove highly correlated with amount max
✓	prices.dateAdded	●		8008	date
	prices.amountMax	●	☆ label var 0%	6526	str → float
✓	keys	●		9963	id-like
✓	dateUpdated	●		7966	date
✓	dateAdded	●		7855	
✓	categories	● ●		1263	csv string with categories → parse, dummyfy
✓	name	●		9634	str with name
✓	id	●		9963	id-like

Variables After Deleting

✓ 1. prices.amountMax 🟣 ← removed illegal values, converted to float

🔴 2. prices.size 🔵 delete since we already have sizes below

🔴 3. prices.color 🔵 drap this too, same reasons

🟢 4. prices.returnPolicy 🔵 ready for categorizing

🟢 5. reviews 🟢🟣 calculated average rating for available reviews

🟢 6. dimension 🟢🟣 converted to number → sum of dims

🟢 7. Prices.shipping 🔵🟢 split into free vs. non-free

🟢 8. sizes 🟢🟣 calculated number of sizes available

🟢 9. prices.offer 🔵

🟢 10. manufacturer 🔵 ready for cat

🟢 11. colors 🔵🟢 split, turned into str, ready to encode / categorize

🟢 12. prices.condition 🔵 ready for categorization

🟢 13. prices.merchant 🔵 ready for categorization

🟢 14. brand 🔵 cleaned, ready to be categorized

🟢 15. prices.currency 🟢🔵 no NaNs, clean, ready to be encoded

🟢 16. prices.isSale 🔵🟢 cleaned, converted to Boolean

🔴 17. categories 🟢🔵 I decided to remove as it is very messy and not informative, I could use NLP embeddings to featurize, but I'll skip for now

can also use TfidVectorizer, but have to deal with nans and will bloat feature space

↓

Processing & Modeling Pipeline

① read in csv file, ignore invalid rows

② remove rows with all NaNs

③ remove rows with NaNs percentage 🔵 > 99%

④ remove cols marked with 🔴 from above

⑤ perform cleaning on remaining 🟢 🔵 🟣 columns

Data Cleaner

custom transformer class

data preparation step

↓

create a single class to do it all