Tubular Data Science Coding Test

answered by Tiecheng Zhou

Note: I'm analyzing the first 60,000 lines from animals_comments.csv, ~ 25,000 users, in order to speed-up my model-fit. The codes should work with the whole dataset.

Step1: Identify Cat And Dog Owners

My code: step1.py

Description:

This code is to identify cat/dog owner based on whether or not the keywords (string) "dog", "pup", "cat", "kitten" are mentioned in the user's comment. If these keywords are mentioned in a comment, this comment will be regarded as an evidence that supports the user being a cat/dog owner.

Results:

This step will produce a "dog_owner" tag and "cat_owner" tag for each userid.

Step2: Build And Evaluate Classifiers

My code: step2.py

Description:

This code is to build cat_owner_classifier, and dog_owner_classifier. I'm using "LogisticRegression" for classification (binary classification, i.e. a user is either being a cat owner or not).

Featue: "all-the-comments", which is a combination of all the comments that a user has made. It will be tokenized using "RegexTokenizer" and transformed using "CountVectorizer".

Label: cat_owner_tag or dog_owner_tag from step1_output.

Cross validation: I'm using brute-force hyperparameter optimizing:

$J(w) = \sum(y - f_w(x)^2 + \alpha(\lambda\|w\|_1) + (1-\alpha)(\frac{\lambda}{2}\|w\|_2^2)$

with $\alpha \in \{0.0, 0.5, 1.0\}$ and $\lambda \in \{0.0, 0.5, 1.0\}$. The best-model for dog classifier has $\{\alpha = 0.5, \lambda = 0.0\}$, while it has $\{\alpha = 0.0, \lambda = 0.0\}$ for cat-classifier.

The performances of the classifiers are measured by two ways: (1) from "BinaryClassificationEvaluator", and (2) from true-positive-rate (TPR) and true-negative-rate (TNR). The best-classifiers are saved for later use.

Results:

The dog_owner_classifier has better performance in terms based on the evaluator.

Performance	Dog_owner_classifier	Cat_owner_classifier
evaluator	0.949505	0.754067
TPR	0.116031	0.599631
TNP	0.998182	0.956761

Step3: Classify All The Users

My code: step3.py

Description:

This code is to estimate all the users who are cat/dog owners (I used 25,000 users). The pipeline and model is read from step2 output.

Fraction: The fraction is calculated as the count of cat/dog owners predicted by the mode divided by the total number_of_users.

Results:

Here I calculate the fraction from the classifier (model from step2), and compare it with that from direct string search (step1). %step2 is smaller than %step1, which is consistent with the small TPR value above.

	Dog-owner users	Cat-owner users
%users_by_step2	2.515 %	7.337 %
%users_by_step1	8.296 %	6.962 %

Step4: Extract Insights About Cat And Dog Owners

My code: step4.py

Description:

I'm selecting keywords with highest coefficients in the model. Based on the "LogisticRegression", i.e. the logistic function: $y = f_w(x) = {}^{1}/_{1 + e^{-b-w^T x}}$, it can be seen that a big weight-coefficient w will make the corresponding x more important to the prediction of y. When "countVectorizer" is used in my model, all the x are positive or zero, therefore the larger w is, the more important x is. Thus, I read the coefficients from the model, find the largest one (or ones) and get the corresponding keyword (it is saved to a log file: log.step4), which are listed below.

Results:

For dog-owners, the keyword "frapupccino" is important in their comments, i.e. dog-owners may like frapupccino.

	Dog_owner_classifier		Cat_owner_classifier	
	Coefficient	keywords	Coefficient	keywords
Most-important	1.575189	"frapupccino"	311.9308	"hampster", "amazons", "fahaka", "mist", "cories", "codepasses", "aquaclear20", "sanding", "501", "notificationstarts", "stratum", "antifungal", "aq20", "hairgrass", "cinematic", "excitementopenssees", "fuhaka", "laguna", "k1", "optimized", "mysis", "probobly", "ramshorn", "aqueon"
Compare	0.358228	"dog"	41.17992	"cat"
	0.431717	"pup"	67.55490	"kitten"

Step5: Identify Creators With Cat And Dog Owners In The Audience

My code: step5.py

Description:

I'm using the prediction of cat/dog owners from step3, and combine it with the dataset. Then I group by the "creator_name", and collect the number_of_users, number_of_cat_owner_users, number_of_dog_owner_users, and the associated fractions. I save these as csv files in step5_xx folders.

Results:

There are many creators that has the highest percentage of cat/dog owners, when there is only 1 audience (here I only list three, the actual list can be folder in the folders)

Top 3 creators that has the most dog_owner audience

Creator_name	N_{dog_owner}	N_{cat_owner}	$N_{audience}$	f_{dog_owner}	f_{cat_owner}
"The Dodo"	170	198	1577	0.1078	0.1256
"Brave Wilderness"	110	395	6678	0.0164	0.0591
"Hope For Paws – Official Rescue Channel"	93	96	935	0.0994	0.1027

Top 3 creators that has the highest percentage of dog_owner audience

Creator_name	N_{dog_owner}	N_{cat_owner}	$N_{audience}$	f_{dog_owner}	f_{cat_owner}
"Ashley Siemon"	1	0	1	1.00	0.00
"Ty The Dog Guy"	1	0	1	1.00	0.00
"Pams Dog Academy"	1	1	1	1.00	0.00

Top 3 creators that has the most cat_owner audience

Creator_name	N_{dog_owner}	N_{cat_owner}	$N_{audience}$	f_{dog_owner}	f_{cat_owner}
"Brave Wilderness"	110	395	6678	0.01647	0.05915
"Robin Seplut"	33	222	433	0.07621	0.51270
"The Dodo"	170	198	1577	0.1078	0.1256

Top 3 creators that has the highest percentage of cat_owner audience

Creator_name	N_{dog_owner}	N_{cat_owner}	$N_{audience}$	f_{dog_owner}	f_{cat_owner}
"D.J. Russ teh Chuck E Cheese Thingy"	0	1	1	0.00	1.00
"ADGVibe"	0	1	1	0.00	1.00
"Robert Giroux"	0	1	1	0.00	1.00