Kaggle Days Meetup Milan #2

Intro

Kaggle PetFinder Competition: Deep Learning for… puppies!
Luca Massaron – Lead Data Scientist @ Cattolica Assicurazioni

Ongoing Kaggle Competitions
Alberto Danese – Head of Data Science @ Nexi

Aperitivo and networking

Local Organizers:

Alberto Danese
Luca Massaron

Main Organizers:

Sponsored by:

Kaggle PetFinder Competition

~~Deep Learning~~ Machine learning
For Puppies

Luca Massaron

Who I am

1. Lead Data Scientist, 15+ years of experience in quantitative roles
2. Author of books on Data Science, Machine Learning, Deep Learning and AI
3. Google Developer Expert in Machine Learning
4. Kaggle Master, highest rank achieved on Kaggle: 7th worldwide (153 competitions: 4 gold medals, 27 silver medals, 36 bronze medals)
5. Successfully operated in different sectors such as telecommunications, oil & gas, new media, insurance & finance, consumer goods, trade fairs, public administration, real estate
6. Lecturer in marketing and statistics at universities and private business schools

Maybe you know me for some books

For Dummies
@ForDummies

Segui

"[Mark] Cuban even said he keeps a "Machine Learning for Dummies" book in his bathroom at home." Pretty sure this is a compliment! @mcuban ow.ly/pttv50uvnsv @lucamassaron

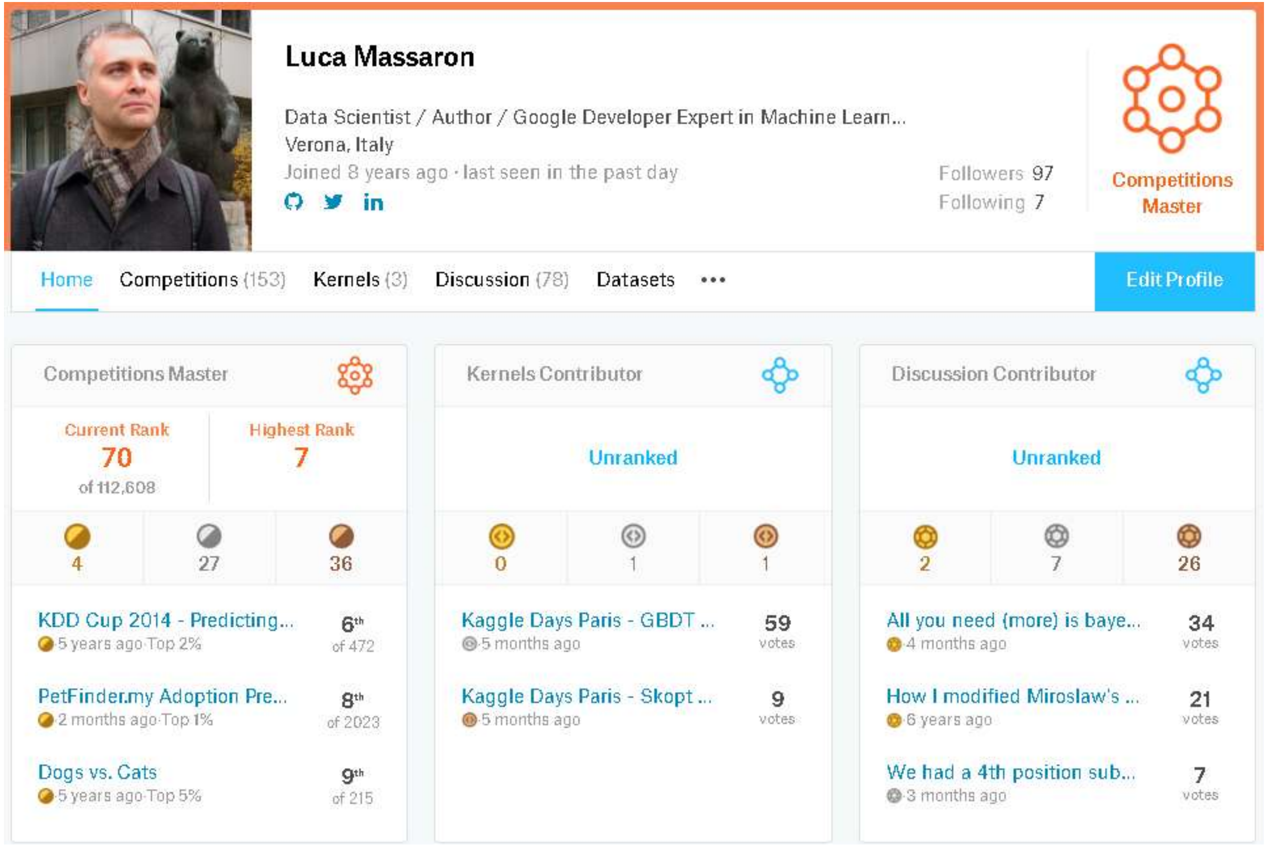
Traduci il Tweet

Why Mark Cuban is taking coding classes

Dallas Mavericks owner Mark Cuban explains why he takes online coding classes.

finance.yahoo.com

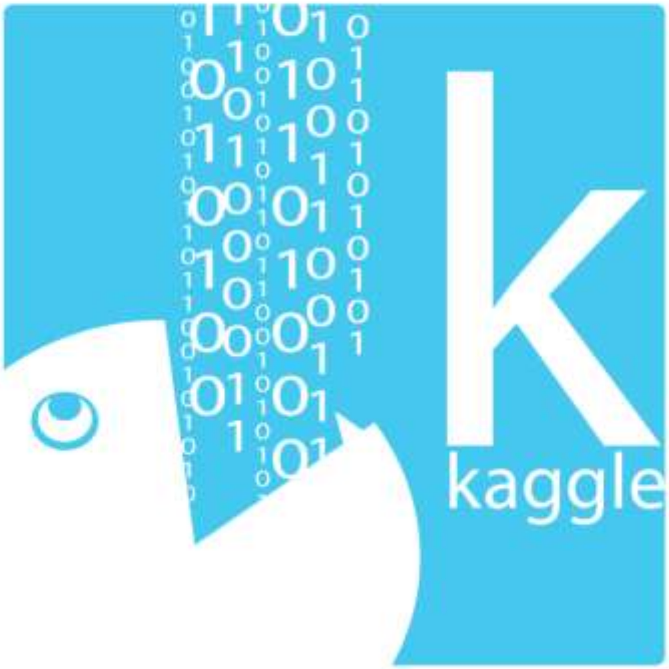
…or for my legacy in Kaggle



kaggle DAYS MEETUP

What is Kaggle?

- Leading platform for machine learning competitions since 2010
- Companies post real data and problems that can be solved with predictive modeling / machine learning / AI / some kind of magic!
- Data scientists from all over the world compete to produce the best algorithms
- Acquired by Google in 2017
- Grown to a complete ML platform with learning modules, code sharing features (kernels), job board and more



LOGICAI kaggle

Of course, real world is not like Kaggle

- First comes the problem, then the data science solution
- You have to study and do a lot of research first
- You have to abide regulations (like GDPR) and licenses
- You have to find the right data, pipeline and prepare it
- You should not snoop at the test data
- You cannot over-engineer your ML solutions because time and resources are stringent constraints

Sources:
https://towardsdatascience.com/data-science-in-the-real-world-e97e2534e43
(**Data Science in the Real World ,**Jan Zawadzki, Data Scientist@Volkswagen Group)

But Kaggle makes data useful

- You have the opportunity to work with data you don’t see at work or at university
- You exclusively work with the latest and most effective techniques (i.e. XGBoost and Keras were launched on Kaggle)
- You can rely on a lot of support from Kaggle for learning (courses, discussion boards, a blog) and computing (they offer you cloud machines)
- You learn transferable skills, even when it doesn’t seem so (i.e. hunting for leakages)

And there's a fantastic community!

PetFinder.my Adoption Prediction

PetFinder.my has been Malaysia's leading animal welfare platform since 2008, with a database of more than 150,000 animals. PetFinder collaborates closely with animal lovers, media, corporations, and global organizations to improve animal welfare.

In this competition you will be developing algorithms to predict the adoptability of pets - specifically, how quickly is a pet adopted? If successful, they will be adapted into AI tools that will guide shelters and rescuers around the world on improving their pet profiles' appeal, reducing animal suffering and euthanization.

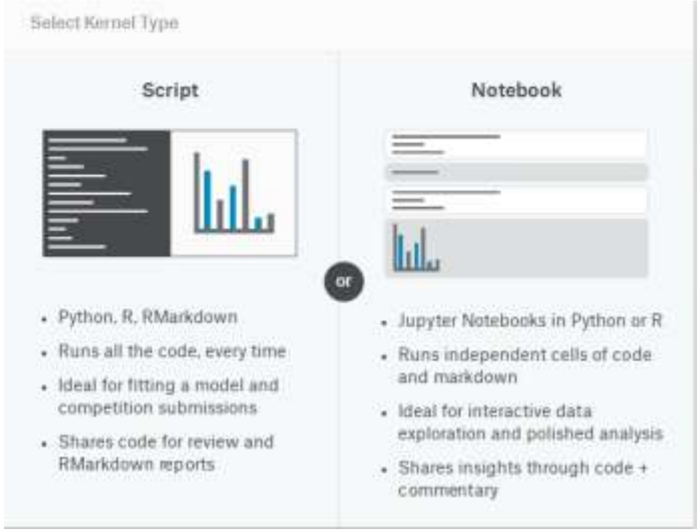
Why it has been so interesting ☺

1. The target is predicting how long it will take for a pet to be adopted, but the problem could be generalized to other social / business domains.

2. The data is interesting because of size (manageable) and because of variety (tabular, text, and image)

3. It was a kernel competition, forcing the participants to mix performance, and reproducibility of results under hardware and time constraints

What is a kernel competition?



1. Limited to either 2 hours on the Kaggle servers with a GPU (Tesla K80) or 6 hours without a GPU.

2. External data was admissible as long as it was not taken from the PetFinder website.

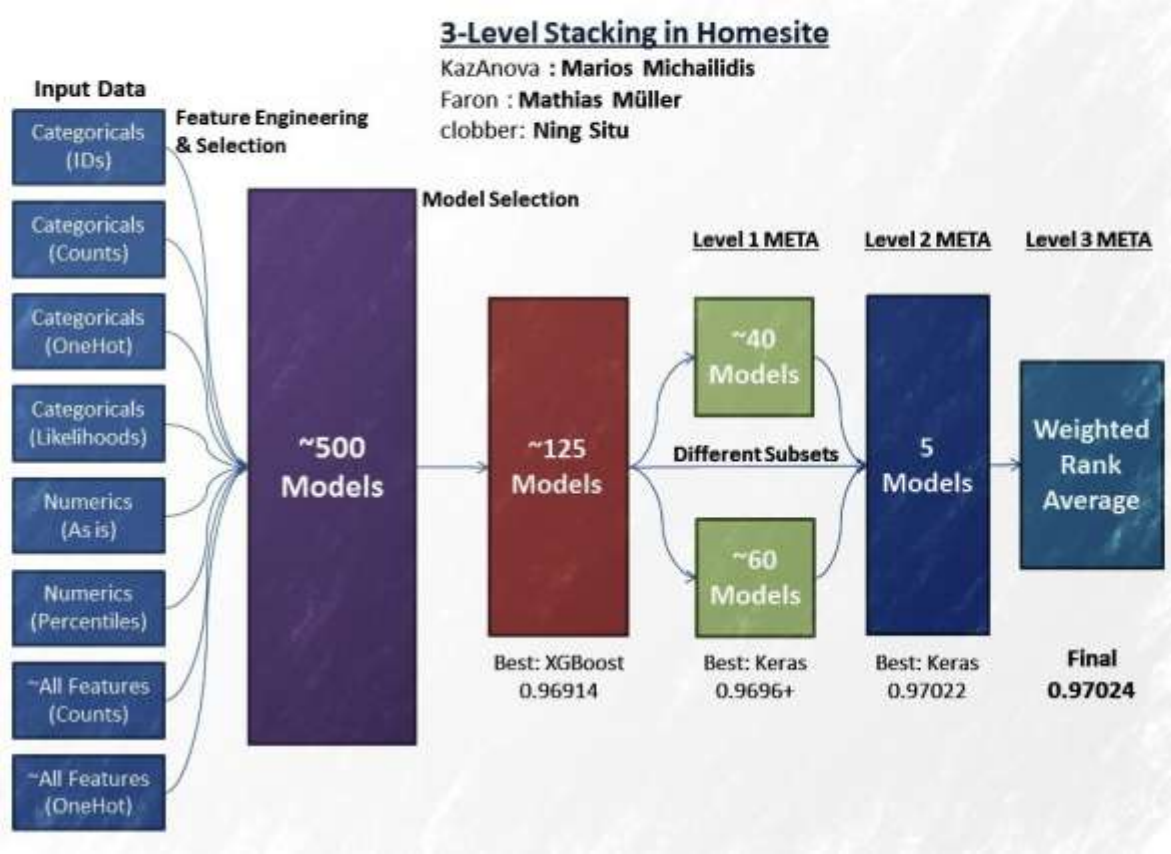
3. The internet must be turned off, so no direct download during inference.

4. No pre-computed predictions or features therefore we had to re-train the models during inference time.

Kaggle Kernels is a cloud computational environment that enables reproducible and collaborative analysis.

See: https://www.quora.com/What-is-a-kernel-in-Kaggle

On kernel comps you cannot do this:



http://blog.kaggle.com/2016/04/08/homesite-quote-conversion-winners-write-up-1st-place-kazanova-faron-clobber/

Let's go back to the comp:

Team Members (6 of 8 maximum)

Luca Massaron (you)	Leader	https://www.linkedin.com/in/lmassaron/
Aditya Soni	Member	https://www.linkedin.com/in/aditya-soni-0505a9124/
Shahebaz	Member	https://www.linkedin.com/in/shaz13/
Sanyam Bhutani	Member	https://www.linkedin.com/in/sanyambhutani/
Rishi Bhalodia	Member	https://www.linkedin.com/in/rkbhalodia927/
Bac Nguyen	Member	https://www.linkedin.com/in/bac-nguyen-xuan-70340b66/

Team name: We Need A Fulltime Job

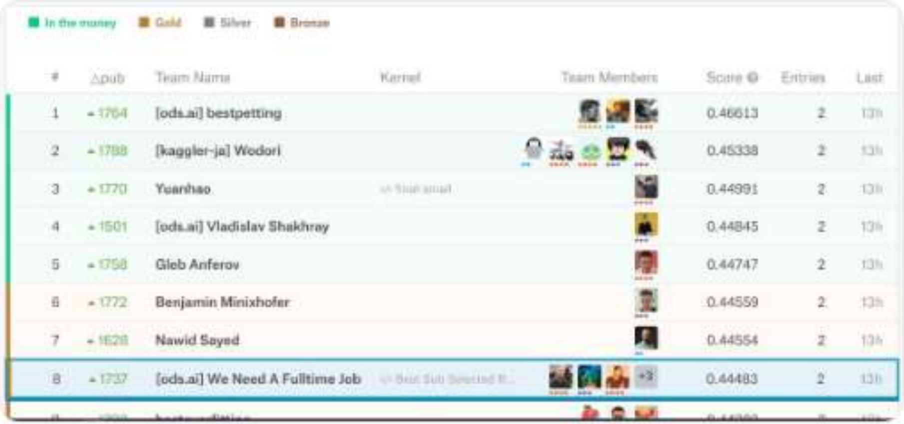
Wondering about the team name?

Sanyam Bhutani @bhutanisanyam1 · 10 apr

Dreams do come true: Our team finished 8th on the @kaggle @myPetFinder Adoption Prediction Challenge. Bringing the first comp category Gold to my profile.

TBH, all credit goes to my amazing teammates @byteshaz, @aditya_soni2k17, Luca Massaron, Bac Ng., @RKBhalodia_927

Traduci il Tweet



In the money · Gold · Silver · Bronze

#	Δpub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲1764	[ods.ai] bestpetting			0.46613	2	13h
2	▲1788	[kaggler-ja] Wodori			0.45338	2	13h
3	▲1770	Yuanhao	[illegible]		0.44991	2	13h
4	▲1501	[ods.ai] Vladislav Shakhray			0.44845	2	13h
5	▲1758	Gleb Anferov			0.44747	2	13h
6	▲1772	Benjamin Minixhofer			0.44559	2	13h
7	▲1628	Nawid Sayed			0.44554	2	13h
8	▲1737	[ods.ai] We Need A Fulltime Job	[illegible]	+3	0.44483	2	13h

22 · 6 · 217

Mostra questa discussione

As you are busy, meanwhile on Twitter…

Sanyam Bhutani @bhutanisanyam1 · 10 apr

PS: Our team name says "We Need A Fulltime Job".
We all are on the job market, DM(s) are open for all 🙂

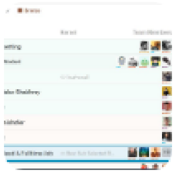
Traduci il Tweet

4 4 27

Mostra questa discussione

Jeremy Howard @jeremyphoward · 10 apr

Wow a whole team of deep learning experts that are on the job market. If you're hiring and want people that actually know how to train accurate models, here's your chance!

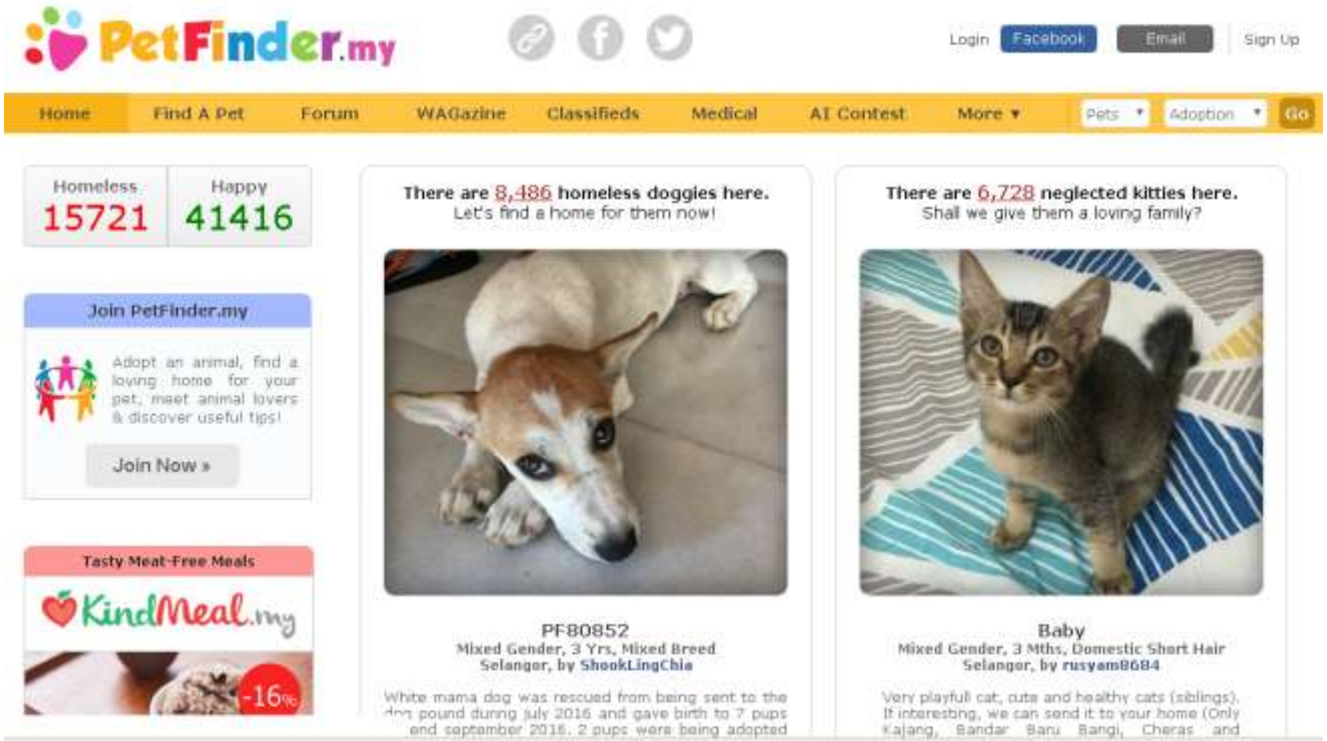


> **Sanyam Bhutani** @bhutanisanyam1
> Dreams do come true: Our team finished 8th on the @kaggle @myPetFinder Adoption Prediction Challenge.
> Bringing the first comp category Gold to my profile.
> TBH, all credit goes to my amazing teammates ...
> Mostra questa discussione

Traduci il Tweet

6 33 214

A look at the Website



You have rankings for pets

Pets For Adoption

Browse By State

- Selangor
- Kuala Lumpur
- Johor
- Kedah
- Kelantan
- Melaka
- Negeri Sembilan
- Pahang
- Perak
- Perlis
- Pulau Pinang
- Sabah
- Sarawak
- Terengganu
- All States

Browse By Pets

- Dog
- Cat
- Rabbit
- Hamster
- Small & Furry
- All Pets

Browse Available Pets

Click on a Pet Profile to view its details and submit enquiry. You can refine your search at the **Advanced Search** section, or **List Your Pet** here.

Page: 1 2 3 4 5 6 7 8 9 10 Next »

Thursday
For Adoption, Selangor
1 Year, Female
Domestic Medium Hair
LadYieyta | Jun 7th

Figaro
Lost in Kuala Lumpur
3 Years, Male
Domestic Medium Hair
Susanna | Jul 3rd

Elvis
Lost in Puchong
8 Years, Male
German Shepherd Dog
ArunKaruppan | Sep 27th

Cheshire
For Adoption, Kuala Lumpur
3 Years, Female
Domestic Short Hair
JueyoungWon | May 26th

Pets are listed

« Back To Listing

Thursday

Cat	Domestic Medium Hair
Profile	Female, 1 Year 2 Months
Vaccinated	Yes
Dewormed	Yes
Spayed	Yes
Condition	Healthy
Body	Medium Size, Medium Fur
Color	Gray
Location	Subang Jaya, Selangor
Posted	31 Dec 2018 (Updated 7 Jun 2019)
Adoption Fee	FREE

Rescuer
LadYleyta

Send Email

View Phone

Write Comment

She is a sweet little catgirl named Thursday. We have actually had her for a month, but we still processing the event surrounding her rescue, which were very tragic. She was limping, nutrient deficiency condition, covered in fleas and had a flu. After vetted, deflead, de-wormed and treatment, she is now healthy, happy, super lovable, playful and incredibly affectionate. We gonna get her vaccinated,and neutered soon. Please open your heart to this pretty girl.

And so are rescurers

Lots of insights to keep account of

Nefeli Kousi
146th place

Stray animals in Malaysia: the Reality I Saw Travelling There For the Past Months

55

posted in PetFinder.my Adoption Prediction 2 months ago

I spent a good two months in Malaysia working as volunteer in marine life conservation projects. In this article I will outline the things I saw and the unique issues that Malaysian strays face.

https://www.kaggle.com/c/petfinder-adoption-prediction/discussion/86581#latest-505147

- The stumpy tailed cats of Malaysia. Are a local breed of cats with short tails, often twisted at the end. A lot of people consider cats with full tails *“cutter”* and prefer them as pets.
- There a cultural / religious reasons for considering not acceptable having a dog as a pet. Dogs have generally a harder life than cats.
- Moreover generally cats seemed to be considered as “pets” while dogs were seen more as “useful”, so people tend to prefer bigger dogs.
- Can you see both eyes in the picture? (a picture of a dog that looks straight in the camera signifies friendliness and good behavior). Is the fur intact? Is the tail shown in the picture?

A glance at the features (1)

- PetID - Unique hash ID of pet profile
- AdoptionSpeed - Categorical speed of adoption. Lower is faster.
  This is the value to predict.
- Type - Type of animal *(1 = Dog, 2 = Cat)*
- Name - Name of pet *(Empty if not named)*
- Age - Age of pet when listed, in months
- Breed1 - Primary breed of pet *(Refer to BreedLabels dictionary)*
- Breed2 - Secondary breed of pet, if pet is of mixed breed
  (Refer to BreedLabels dictionary)
- Gender - Gender of pet *(1 = Male, 2 = Female, 3 = Mixed, if profile represents group of pets)*
- Color1 - Color 1 of pet *(Refer to ColorLabels dictionary)*
- Color2 - Color 2 of pet *(Refer to ColorLabels dictionary)*
- Color3 - Color 3 of pet *(Refer to ColorLabels dictionary)*
- MaturitySize - Size at maturity *(1 = Small, 2 = Medium, 3 = Large, 4 = Extra Large, 0 = Not Specified)*
- FurLength - Fur length *(1 = Short, 2 = Medium, 3 = Long, 0 = Not Specified)*

A glance at the features (2)

- Vaccinated - Pet has been vaccinated *(1 = Yes, 2 = No, 3 = Not Sure)*
- Dewormed - Pet has been dewormed *(1 = Yes, 2 = No, 3 = Not Sure)*
- Sterilized - Pet has been spayed / neutered *(1 = Yes, 2 = No, 3 = Not Sure)*
- Health - Health Condition *(1 = Healthy, 2 = Minor Injury, 3 = Serious Injury, 0 = Not Specified)*
- Quantity - Number of pets represented in profile
- Fee - Adoption fee *(0 = Free)*
- State - State location in Malaysia *(Refer to StateLabels dictionary)*
- RescuerID - Unique hash ID of rescuer
- VideoAmt - Total uploaded videos for this pet
- PhotoAmt - Total uploaded photos for this pet
- Description - Profile write-up for this pet. The primary language used is English, with some in Malay or Chinese.

A glance at the texts

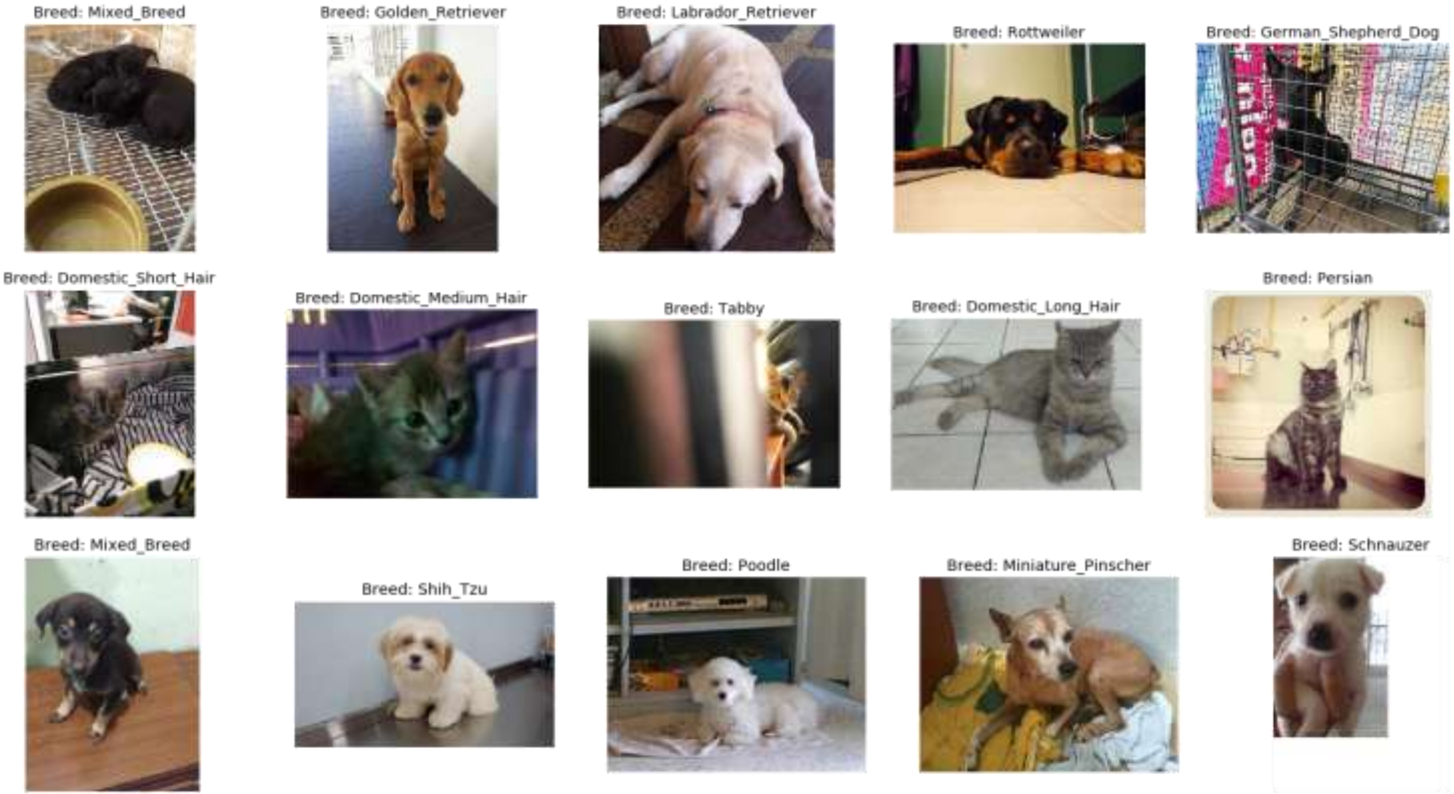
The text is contained in json files processed by Google API:

{ "text": { "content": "Cherry loves to be indoor, loves to be near human, loves human touches, loves other dogs.", "beginOffset": -1 }, "sentiment": { "magnitude": 0.8, "score": 0.8 } },

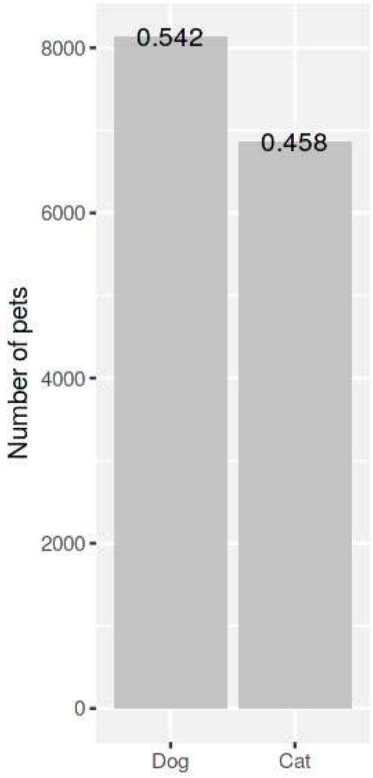
{ "text": { "content": "Don't be mistaken, Cherry is very alert at strangers and noises at the gate, but being a watch dog should not be her full time 'job'.", "beginOffset": -1 }, "sentiment": { "magnitude": 0.5, "score": -0.5 } },

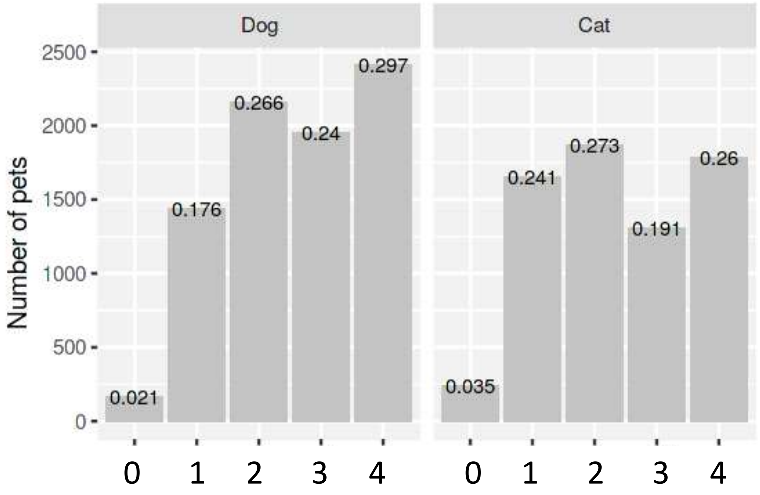
"tokens": [], "entities": [{ "name": "Cherry", "type": "PERSON", "metadata": {}, "salience": 0.703432, "mentions": [{ "text": { "content": "Cherry", "beginOffset": -1 }, "type": "PROPER" },

A glance at the pictures



A glance at the target



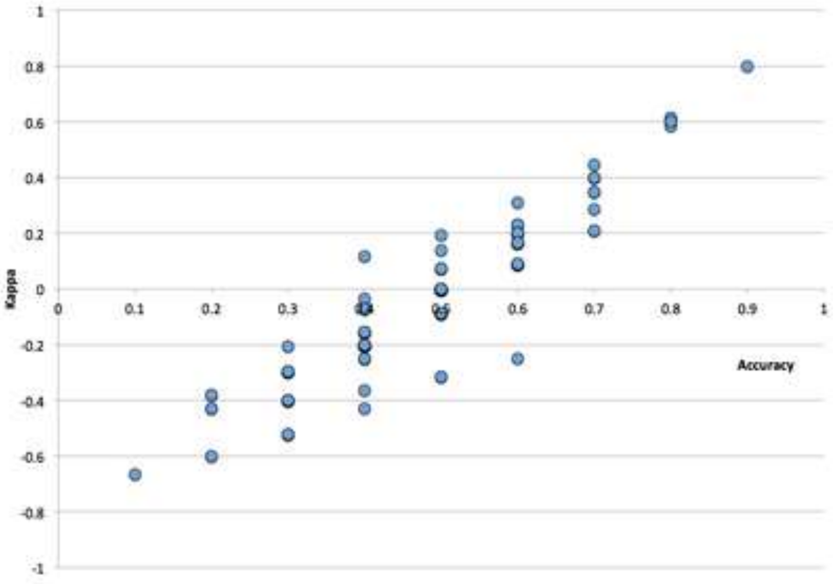


- 0 -Pet was adopted on the same day as it was listed.
- 1 - Pet was adopted between 1 and 7 days (1st week) after being listed.
- 2- Pet was adopted between 8 and 30 days (1st month) after being listed.
- 3 - Pet was adopted between 31 and 90 days (2nd & 3rd month) after being listed.
- 4 - No adoption after 100 days of being listed. (There are no pets in this dataset that waited between 90 and 100 days).

Evaluation function

Submissions are scored based on the **quadratic weighted kappa**, which measures the agreement between two ratings.

This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters). In the event that there is less agreement between the raters than expected by chance, the metric may go below 0.




```
from sklearn.metrics import cohen_kappa_score

def quadratic_weighted_kappa (y_true, y_pred):
    return cohen_kappa_score(y_true, y_pred, weights='quadratic')
```

Our optimization

Solution: we optimize first for rmse, thus handling the problem as a regression problem, then we tune the solution using the out of fold predictions in order to set a numeric threshold and correctly guess the 5 classes and maximize the **quadratic weighted kappa**

```
class OptimizedRounder(object):

…
def fit(self, X, y):
        loss_partial = partial(self._kappa_loss, X=X, y=y)
        initial_coef = [0.5, 1.5, 2.5, 3.5]
        self.coef_ = sp.optimize.minimize(loss_partial,  initial_coef,
                                          method='nelder-mead')
```

* functools.partial returns the inputted function with predefined positional parameters

The data pipeline

train test

Loading train, test, breed, color, state labels

load_tabular_data()

TRAIN: (14993, 24)
Wall time: 230 ms

Feature engineering aiming at transformng, grouping, averaging features

basic_features()

TRAIN: (14993, 396)
Wall time: 1min 6s

Basic text features such as length, number of words, smileys

meta_nlp_feats()

TRAIN: (14993, 408))
Wall time: 2.51 s

processing Google's Vision API general image data present on -1.json files

bounding_features()

TRAIN: (14993, 418)
Wall time: 43 s

processing all metadata fromGoogle's Vision API and Google's Natural Language API

metadata_features()

TRAIN: (14993, 447)
Wall time: 24min 40s

... ...

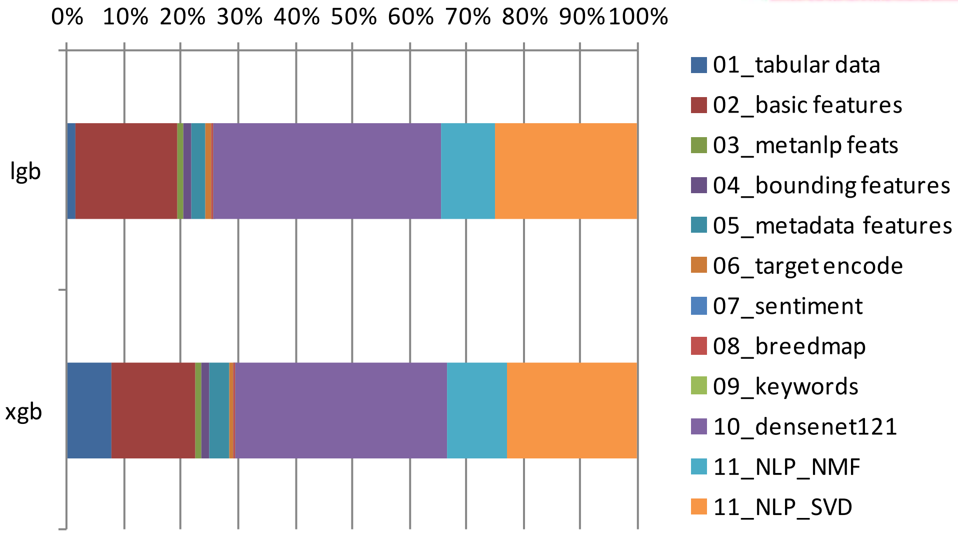
The data pipeline

train → test

Target encoding of key variables	target_encode (Breed1, Breed2, Age)	TRAIN: (14993, 450) Wall time: 255 ms
Each pet is matched with a score, magnitude, negative score	sentiment_analysis	TRAIN: (14993, 453) Wall time: 13.3 s
We create indicator variables for textual keywords and join external data with breeds stats	breed_maps() keyword mapping	TRAIN: (14993, 458) Wall time: 759 ms
Using Imagenet pretrained denseNet121, we extract 256 image features	image_feature(denseNet121)	TRAIN: (14993, 714) Wall time: 14min 57s
Extracting 50 components from SVD and NMF applied to all the different texts available	nlp_features()	TRAIN: (14993, 1009) Wall time: 1min 7s
Adoption speed by blending solutions from XGBoost and LightGBM	(run_lgbm + run_xgb) / 2	Wall time: 53min 7s

Importance of feature blocks

Feature block	n. features	Lgb splits	Lgb gain	Xgb splits
01_tabular data	19	1,40%	3,33%	7,73%
02_basic features	372	17,88%	28,82%	14,66%
03_metanlp feats	12	1,07%	0,93%	1,06%
04_bounding features	10	1,33%	2,12%	1,47%
05_metadata features	27	2,65%	2,47%	3,60%
06_target encode	3	0,83%	1,46%	0,76%
07_sentiment	3	0,02%	0,02%	0,01%
08_breedmap	4	0,32%	2,44%	0,10%
09_keywords	1	0,00%	0,00%	0,01%
10_densenet121	256	40,15%	32,60%	37,20%
11_NLP_NMF	150	9,50%	7,24%	10,51%
11_NLP_SVD	150	24,85%	18,58%	22,90%
	1007	100,0%	100,0%	100,0%



For comparison reasons we mostly used the importance given by the number of splits that involved a feature in the iterations of the GBMs.

However, splits don’t tell all the story, since features with less levels may need lto be less splitted or simply a single split may hid a huge gain in the cost function.

A few simple key ingredients

1. Handmade feature engineering together with some business understanding

2. Target encoding (reduce cardinality)

3. A pre-trained network, denseNet121, for generating features from images

4. NLP by SVD (LSA) and NMF (Topic Modelling)

5. Two power horses such as XGBoost and LightGBM

6. Solving it as a regression problem, on a linear continuum, then optimally discretized accordingly to the competition's metrics

(But remember that "There is no free lunch")

basic_features

First we apply basic feature engineering for better separablity by:

1. Transforming variables
   weeks, Feature_SecondaryColors, Feature_MonoColor, total_img_video,

2. Grouping/clustering variables
   L_Breed1_Siamese, L_Breed1_Persian, L_Breed1_Labrador_Retriever, L_Breed1_Terrier, shorthair_hairless_domestic_hair, top_dogs, top_cats

3. Taking averages of groups
   Feature_avg_age_breed1_fee, Feature_age_breed1_maturity_sz, Feature_age_breed1_fur, Feature_state_breed1_age_freq, Feature_avg_type_age_breed1_fee, Feature_age_type_breed1_fur …

4. Taking more complex stats
   RescuerID, State expressed as nunique, mean, **var**, **max**, **min**, **skew**, **median of many variables**

5. Ranking (by **seo_value** a proxy based on photo & video) inside the group
   State, Animal, Type, Breed1, Gender

Seo value

```
def seo_value(cols):
        photos = cols[0]
        videos = cols[1]
        seo = .7 * videos + .3 * photos
        return seo

alldata['InstaFeature'] = alldata[['PhotoAmt', 'VideoAmt']].apply(seo_value, axis=1)

def rankbyG(alldata, group):
        rank_telemetry = pd.DataFrame()
        for unit in (alldata[group].unique()):
            tf = alldata[alldata[group] == unit][['PetID', 'InstaFeature', group]]
            col_name = "Insta" + str(group).title() + "Rank"
            tf[col_name] = tf['InstaFeature'].rank(method='max')
            rank_telemetry = pd.concat([rank_telemetry, tf[['PetID', col_name]]])
            del tf
        alldata = pd.merge(alldata, rank_telemetry, on=['PetID'], how='left')
        return alldata
```

Photos and videos availability were treated as a proxy of ranking in the website to use in order to rank within different groups

Target encoding

High cardinality variables are processed using an encoding function which is computed accordingly to the following paper by Daniele Micci-Barreca:

Micci-Barreca, Daniele. "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems." ***ACM SIGKDD Explorations Newsletter*** **3.1 (2001): 27-32**.

Code we used in the competition:
https://gist.github.com/lmassaron/6695171ff45bae7ef7ddcdad2ad493ca

Original version by Olivier Grellier (H2O.ai)
https://www.kaggle.com/ogrellier/python-target-encoding-for-categorical-features

Inputs:

trn_series : training categorical feature as a pd.Series
tst_series : test categorical feature as a pd.Series
target : target data as a pd.Series
min_samples_leaf (int) : minimum samples to take category average into account
smoothing (int) : smoothing effect to balance categorical average vs prior

The idea behind target encoding

$$X_i \rightarrow S_i \cong P(Y|X=X_i) \qquad S_i = \frac{n_{iY}}{n_i}$$

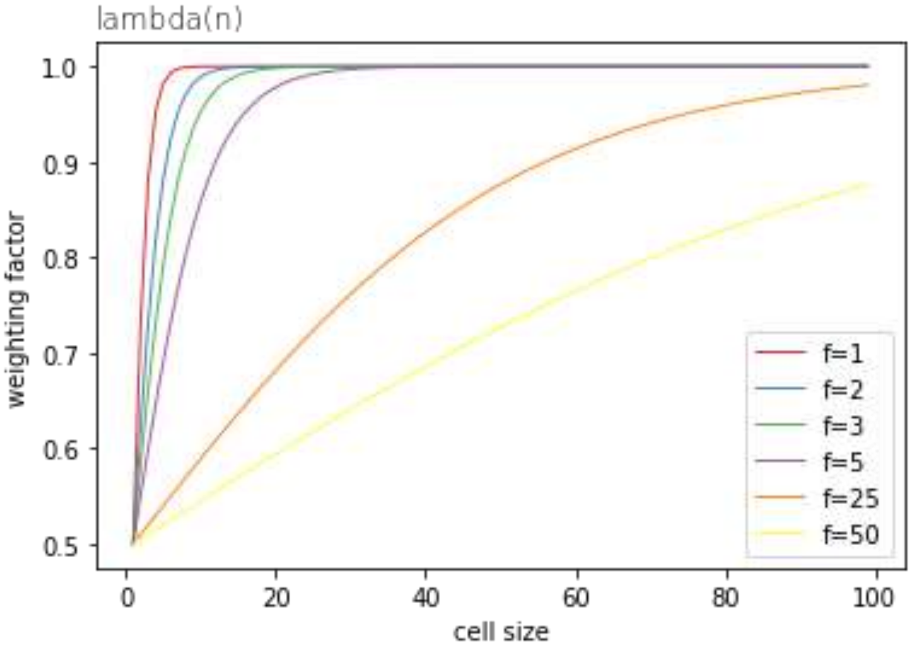
For a level i, we are looking for an approximate value that can help us predict better the target using a single encoded variable. Replacing the level by the observed conditional probability could be the solution, but for the levels with few observations.

$$S_i = \lambda(n_i)\frac{n_{iY}}{n_i} + (1-\lambda(n_i))\frac{n_Y}{n_{TR}}$$

The solution is to blend the observed posterior probability on that level (probability of Y given X=Xi) with the a-priori probability (probability of Y) by a lambda factor (Empirical Bayesian approach).

The idea behind target encoding

$$\lambda(n) = \frac{1}{1+e^{-\frac{(n-k)}{f}}}$$



Given a fix k (usually it is 1, implying a minimum cell frequency of 2), higher values of f dictate less trust in the observed empirical frequency and more reliance on the empirical probability for all cells.

denseNet121

DenseNet is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each *dense block*). For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers.

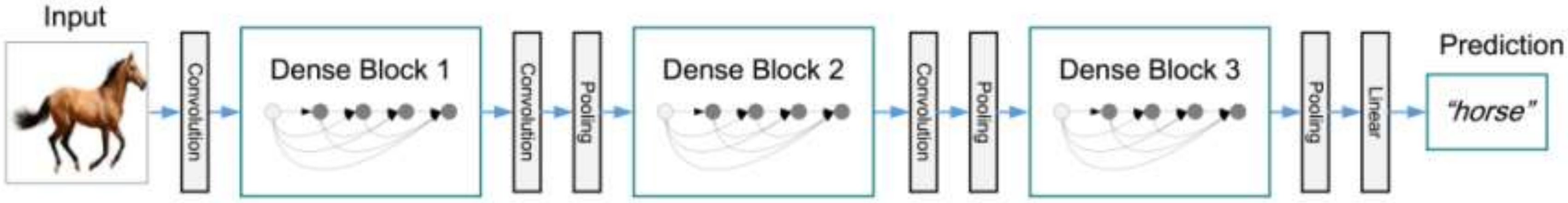
This connectivity pattern yields state-of-the-art accuracies on **CIFAR10/100** (with or without data augmentation) and **SVHN**. On the large scale ILSVRC 2012 (ImageNet) dataset, DenseNet achieves a similar accuracy as ResNet, but using less than half the amount of parameters and roughly half the number of FLOPs.

Source : https://github.com/liuzhuang13/DenseNet

Extracting features from denseNet121

```
def build_model(shape=(256, 256, 3),
  weights_path="../input/densenet-keras/DenseNet-BC-121-32-no-top.h5"):
    inp = Input(shape)
    backbone = DenseNet121(input_tensor=inp,
                                               weights=weights_path,
                                               include_top=False)
    x = backbone.output
    x = GlobalAveragePooling2D()(x)
    x = Lambda(lambda x: K.expand_dims(x, axis=-1))(x)
    x = AveragePooling1D(4)(x)
    out = Lambda(lambda x: x[:, :, 0])(x)
    model = Model(inp, out)
    return model
```

Glancing at the architecture




```
...
__________________________________________________________________________________________________
bn (BatchNormalization) (None, 8, 8, 1024) 4096 conv5_block16_concat[0][0]
__________________________________________________________________________________________________
relu (Activation) (None, 8, 8, 1024) 0 bn[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None, 1024) 0 relu[0][0]
__________________________________________________________________________________________________
lambda_1 (Lambda) (None, 1024, 1) 0 global_average_pooling2d_1[0][0]
__________________________________________________________________________________________________
average_pooling1d_1 (AveragePoo (None, 256, 1) 0 lambda_1[0][0]
__________________________________________________________________________________________________
lambda_2 (Lambda) (None, 256) 0 average_pooling1d_1[0][0]
==================================================================================================
Total params: 7,037,504 Trainable params: 6,953,856 Non-trainable params: 83,648
```

NLP processing

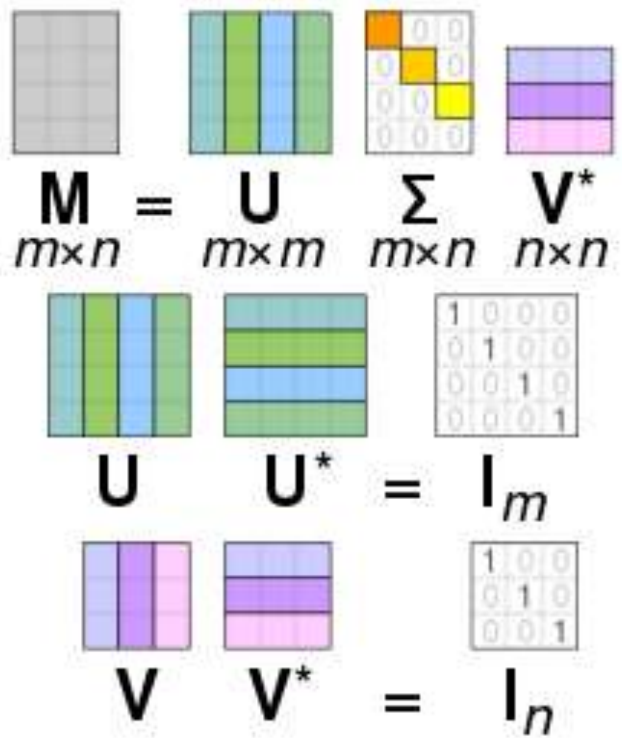
Basic indicator extracted from textual information:

1. Length
2. Capitals
3. caps_vs_length
4. num_exclamation_marks
5. num_question_marks
6. num_punctuation
7. num_symbols
8. num_words
9. num_unique_words
10. words_vs_unique
11. num_smilies (**':-)'**, **':)'**, **';-)'**, **';)'**)
12. num_sad (**':-<'**, **':()'**, **';-()'**, **';('**)))

We couldn't use embeddings or even BERT because of the competition constraints and because as many descriptions were in English, some were also in Malay and Chinese (and we noticed that adoption speed drops for these two languages)

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD, NMF
n_components = 50
svd_ = TruncatedSVD(n_components=n_components, random_state=1337)
nmf_ = NMF(n_components=n_components, random_state=1337)
tfidf_col = TfidfVectorizer().fit_transform(X_text.loc[:, i].values)
svd_col = svd_.fit_transform(tfidf_col)
nmf_col = nmf_.fit_transform(tfidf_col)
```

SVD (LSA)



Latent semantic analysis (LSA) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.

- data clustering, document classification
- cross language retrieval
- synonymy and polysemy
- information retrieval
- expand the feature space of machine learning / text mining systems

SOURCE:
https://en.wikipedia.org/wiki/Singular_value_decomposition
https://en.wikipedia.org/wiki/Latent_semantic_analysis

NMF for topic modelling

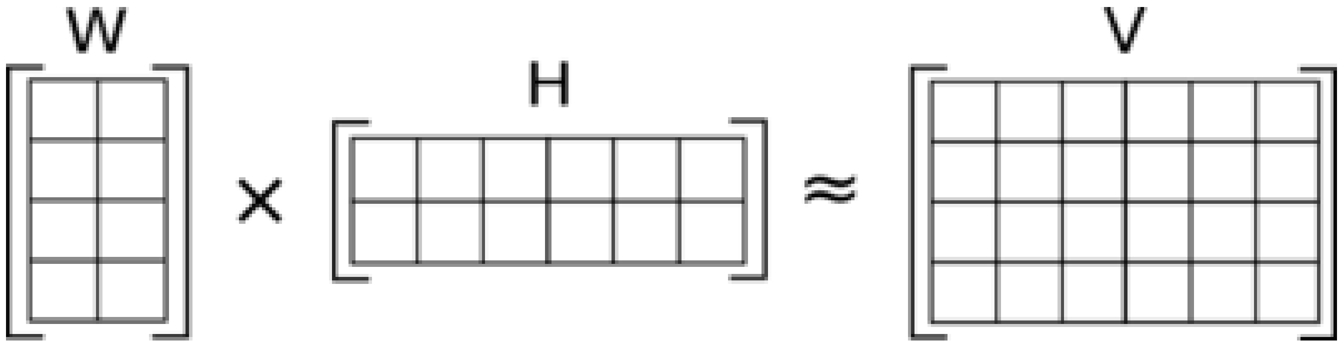


Illustration of approximate non-negative matrix factorization: the matrix **V** is represented by the two smaller matrices **W** and **H**, which, when multiplied, approximately reconstruct **V**.

NMF can be used for text mining applications. In this process, a document-term matrix is constructed with the weights of various terms (typically weighted word frequency information) from a set of documents. This matrix is factored into a term-feature and a feature-document matrix. The features are derived from the contents of the documents, and **the feature-document matrix describes data clusters of related documents**.

SOURCE: https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

XGBoost

XGBoost stands for eXtreme Gradient Boosting, an open source project by Tianqui Chen, Tong He, and Carlos Guestrin that has gained gained momentum and popularity in datascience competitions such as Kaggle and the KDD-cup 2015.

1. **level-wise** and **leaf-wise** splitting strategies (The level-wise strategy maintains a balanced tree, whereas the leaf-wise strategy splits the leaf that reduces the loss the most.)
2. **Weighted Quantile Sketch** for determining how to make splits in a decision tree (candidate splits)
3. **Sparsity-aware Split** finding introduces a default direction in each tree node, so when some data is missing, the direction of the split is automatically predermined, thus reducing complexity row-wise

```
params = {'eta': 0.0123, 'subsample': 0.7, 'colsample_bytree': 0.75,
                    'tree_method': 'gpu_hist', 'gamma' : 8, 'max_depth' : 7}
early_stop = 500
num_rounds = 10000
```

LightGBM

The high-performance LightGBM algorithm is capable of being distributed and of fast-handling large amounts of data. It has been developed by a team at Microsoft as an open source project on GitHub (there is also an academic paper).

1. **leaf-wise** splitting strategy
2. **Gradient-based One-Side Sampling (GOSS)** which inspects the most informative samples while skipping the less informative samples
3. **Exclusive Feature Bundling** which takes advantage of sparse datasets by grouping features in a near lossless way (basically it combines similar columns)

```
params = {'num_leaves': 70, 'max_depth': 9, 'learning_rate': 0.01,
          'bagging_fraction': 0.6, 'feature_fraction': 0.6, 'min_split_gain': 0.02,
          'min_child_samples': 150, 'min_child_weight': 0.02,
          'lambda_l2': 0.0475}
early_stop = 500
num_rounds = 10000
```

CV strategy

```
n_splits = 10
kfold = StratifiedKFold(n_splits=n_splits, random_state=1337)
rescuer_gb_mean = (X_train.groupby('RescuerID')['AdoptionSpeed']
                                                      .agg("mean").reset_index())
rescuer_as_mean = rescuer_gb_mean['AdoptionSpeed_mean'].values

for train_index, valid_index in kfold.split(rescuer_ids, rescuer_as_mean.astype(np.int)):
        rescuser_train_ids = rescuer_ids[train_index]
        rescuser_valid_ids = rescuer_ids[valid_index]
        # Train model, predict on fold, predict on test
        # Store fold predictions in oof, stack test predictions
```

The rescurer in the test set are unknown in the train set. We cannot rely on such information, yet we can stratify by the information that the rescurer carry associated with them, the adoption speed, and replicate such a distribution in our predictions.

The result are the average (corr=0.943) of the averaged predictions of two models, an XGBoost and a LightGBM, trained on 10 cv folds.

Conclusions

In the money | Gold | Silver | Bronze

#	∆pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲1764	[ods.ai] bestpetting			0.46613	2	2mo
2	▲1788	[kaggler-ja] Wodori			0.45338	2	2mo
3	▲1770	Yuanhao	final-small		0.44991	2	2mo
4	▲1501	[ods.ai] Vladislav Shakhray			0.44845	2	2mo
5	▲1758	Gleb Anferov			0.44747	2	2mo
6	▲1772	Benjamin Minixhofer			0.44559	2	2mo
7	▲1628	Nawid Sayed			0.44554	2	2mo
8	▲1737	[ods.ai]We Need A Fulltime Job	Best Sub Selected R...	+3	0.44483	2	2mo
9	▲1722	bestoverfitting			0.44303	2	2mo
10	▲1527	Shakeup is all you need	final_submit_two	+5	0.44296	2	2mo

8th Place Solution Code

Python script using data from multiple data sources · 875 views · 2mo ago · multiple data sources

https://www.kaggle.com/adityaecdrid/8th-place-solution-code

Ask me any question!

Thank you ☺

https://www.kaggle.com/lucamassaron

Ongoing Kaggle Competitions

Alberto Danese

Active competitions overview

Competition	Prize / Teams
Two Sigma: Using News to Predict Stock Movements Use news analytics to predict stock price performance Featured · Kernels Competition · a month to go · news agencies, time series, finance, money	$100,000 2,927 teams
Jigsaw Unintended Bias in Toxicity Classification Detect toxicity across a diverse range of conversations Featured · Kernels Competition · 17 days to go · nlp, biases, text data	$65,000 2,663 teams
Predicting Molecular Properties Can you measure the magnetic interactions between a pair of atoms? Featured · 3 months to go · tabular data, chemistry, regression	$30,000 659 teams
Open Images 2019 - Object Detection Detect objects in varied and complex images Research · 4 months to go · image processing, image data	$25,000 39 teams
Open Images 2019 - Visual Relationship Detect pairs of objects in particular relationships Research · 4 months to go · image data, image processing	$25,000 26 teams
Data Science for Good: City of Los Angeles Help the City of Los Angeles to structure and analyze its job descriptions [illegible] · [illegible] days to go · image data, text data, employment, nlp	$15,000
Instant Gratification A synchronous Kernels-only competition Featured · Kernels Competition · 11 days to go · binary classification, tabular data	$5,000 1,341 teams

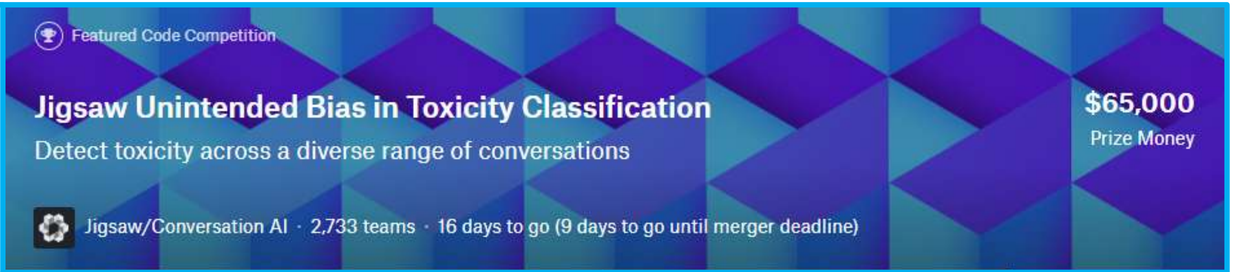
Jigsaw Unintended Bias in Toxicity Classification: (1/2)

Abstract

Can you help detect toxic comments — and minimize unintended model bias?
Toxicity is defined as anything rude, disrespectful or otherwise likely to make someone leave a discussion

Sponsor

Jigsaw and Google (both part of Alphabet)



Prize

65.000$ (12.000$ for 1st place, till 5.000$ for 10th place)

Deadline

26 June

Type of competition

NLP Data

Link

https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

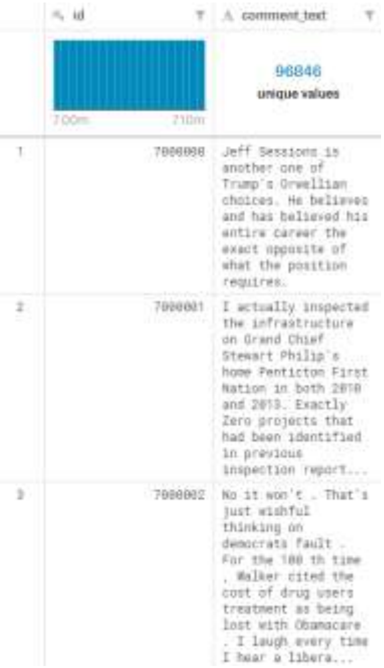
Jigsaw Unintended Bias in Toxicity Classification: (2/2)

Target
Binary (toxic comment or not)

Evaluation metric
Weighted function (AUC based) to take into account the "unintended bias factor"

Data sample
Train: 1.780.000 records
Test: 97.000 records

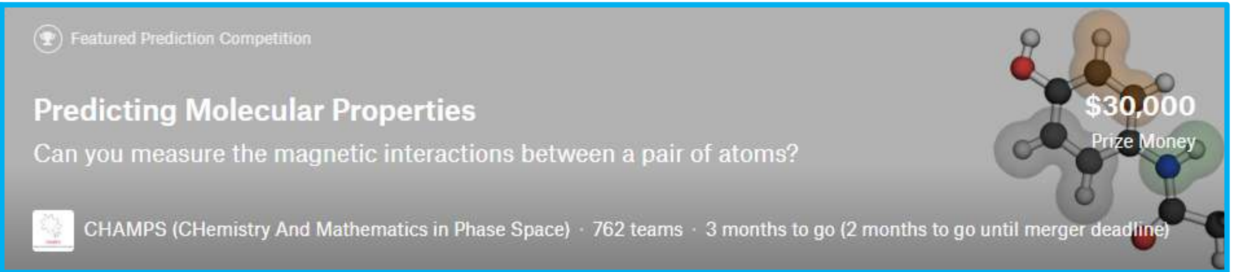
Less than 300 MB total



Predicting molecular properties (1/2)

Abstract

This challenge aims to predict the magnetic interaction between two atoms in a molecule.



Sponsor

Champs (university consortium)

CHAMPS

Prize

30.000$ (12.500$ for 1st place, till 2.000$ for 5th place)

Deadline

28 August

Type of competition

Tabular data (complex structure)

Link

https://www.kaggle.com/c/champs-scalar-coupling

Predicting molecular properties (2/2)

Target

A continuous measure of the interaction (scalar coupling constant)

Evaluation metric

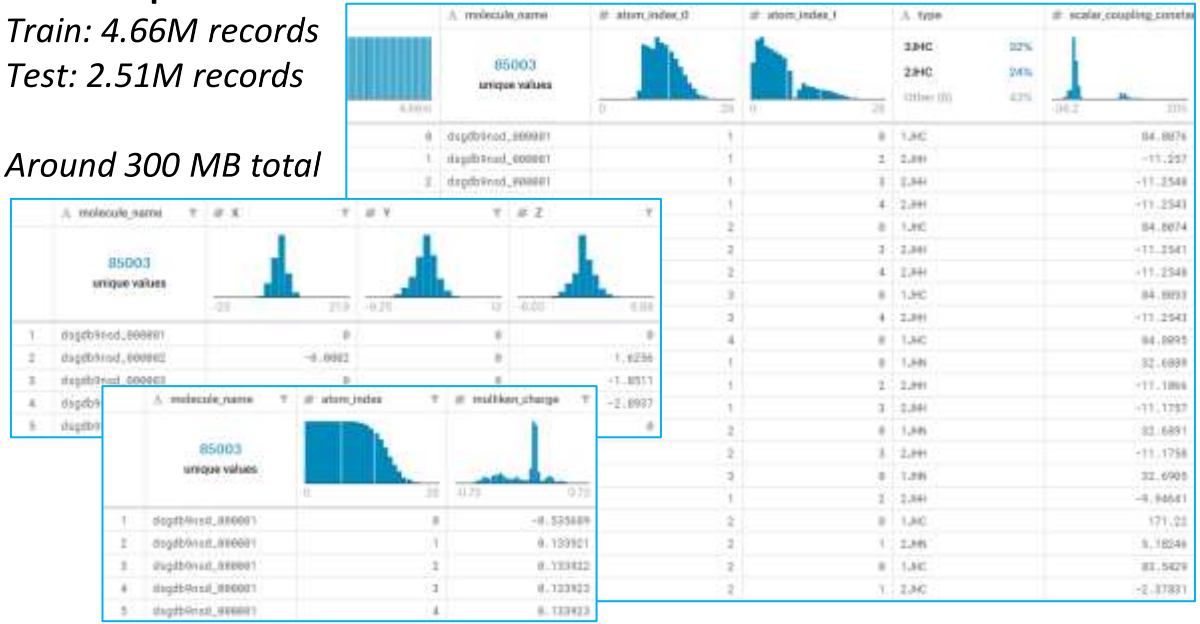
Log of the MAE (mean absolute error) averaged across multiple "coupling types"

Data sample

Train: 4.66M records

Test: 2.51M records

Around 300 MB total

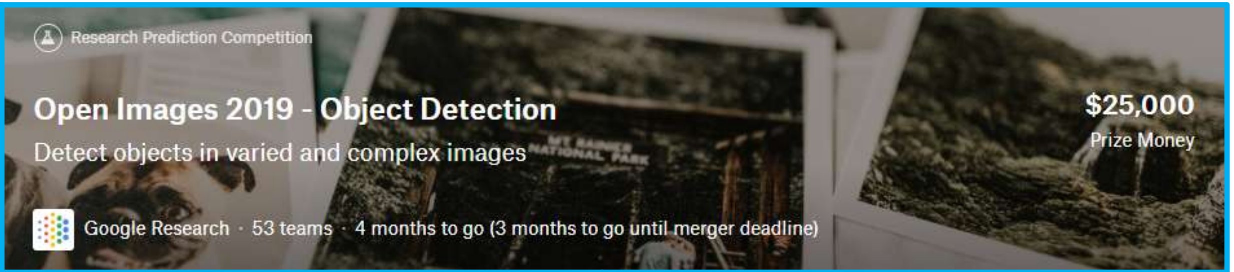


Open Images 2019 – Object Detection (1/2)

Abstract
Computer vision has advanced considerably but is still challenged in matching the precision of human perception. Objective of the challenge is detecting bounding boxes around object instances

Sponsor
Google AI Research

Google AI



Prize
25.000$ (7.000$ for 1^{st} place, till 3.000$ for 5^{th} place)

Deadline
1 October

Type of competition
Image classification

Link
https://www.kaggle.com/c/open-images-2019-object-detection

kaggle DAYS MEETUP

LOGICAI kaggle

Open Images 2019 – Object Detection (2/2)

Target

Sample submission:
ImageID,PredictionString
ImageID,{Label Confidence XMin YMin XMax YMax},{...}

Evaluation metric

Mean Average Precision (over 500 object classes)

Data sample

Train: 1.9M images
Around 561GB

Test: 100K images
Around 10GB

The training set contains 12.2M bounding-boxes across 500 categories on 1.7M images. The boxes have been largely manually drawn by professional annotators to ensure accuracy and consistency. The images are very diverse and often contain complex scenes with several objects (7 per image on average).

Two Sigma: using news to predict stock movements (1/2)

Abstract
Can we use the content of news analytics to predict stock price performance?

Sponsor
New York City based hedge fund focused on AI & ML applied to trading

Prize
100.000$ (25.000$ for 1^{st} place, till 10.000$ for 7^{th} place)

Deadline
15 July (note: new participants no longer allowed)

Type of competition
NLP + Tabular Data

Link
https://www.kaggle.com/c/two-sigma-financial-news/overview/timeline

Two Sigma: using news to predict stock movements (2/2)

Target *A confidence value* $\in[-1,1]$ *,* *which is multiplied by the market-adjusted return of a given* assetCode *over a ten day window.* *Sample submission:* *time,assetCode,confidenceValue* *2019-01-03,RPXC.O,0.1* *2019-01-04,RPXC.O,0.02* **Evaluation metric** *Custom (refer to competition page)*	**Data sample** *Competition in the last phase, will be available for download after the deadline* ***1. Market data (2007 to present)*** *provided by Intrinio - contains financial market information such as opening price, closing price, trading volume, calculated returns, etc.* ***2. News data (2007 to present)*** *Source: Thomson Reuters - contains information about news articles/alerts published about assets, such as article details, sentiment, and other commentary.*

Join us!

Currently looking for:

- *One more organizer*
- *Speakers for next events (from September)*

Alberto Danese – alberto.danese@gmail.com
Luca Massaron – lucamassaron@gmail.com

Or meetup.com, Linkedin… get in touch!