Machine Learning Engineer Nanodegree
Capstone Project

Benjamin Rouillé d'Orfeuil
May 26, 2018

I. Definition

Project Overview

Yelp is a social networking site that publishes crowd-sourced reviews about local businesses. About two years ago, Yelp challenged Machine Learning practitioners to build a model that automatically tags restaurants with multiple labels using a dataset of user-submitted photographs. The goal of this project is to develop such a model.

The competition was hosted by Kaggle, a platform where data scientists use their skills to produce the best models for predicting and describing datasets uploaded by companies and users. The Yelp Restaurant Photo Classification competition can be found at this url:

`https://www.kaggle.com/c/yelp-restaurant-photo-classification`

Problem Statement

Back in Spring 2016, when this competition took place, Yelp users were able to upload photographs and write reviews without having to tag the venue with labels. It follows that some restaurants can be left un- or only partially-categorized. There are lot of photographs uploaded on the Yelp site and, for this reason, business labels need to be predicted by a machine learning model if those have not been selected by users during the submission process.

Each photograph belongs to a business and the task is to predict the business attributes purely from the business photographs. Note that this is a multi-instance multi-label classification problem. Indeed, each business has multiple photographs and predictions need to be done at the business level. Likewise, multiple labels can be assigned to the same business and, as a result, potential dependencies among labels need to be accounted for by the classifier. Classifying real world images is a complex endeavor. This task is often best performed using deep neural networks.

Metrics

The harmonic mean between precision (p) and recall (r), the F_1 score, is used to evaluate the performance of the algorithm:

$$F_1 = 2\frac{p \cdot r}{p + r} \text{ where } p = \frac{tp}{tp + fp} \text{ and } r = \frac{tp}{tp + fn}$$

In the above formula, tp, fp and fn denote the true positive, false positive and false negative counts, respectively. In a classification task, $p = 1$ for class i means that every item labeled as belonging to class i does indeed belong to class i whereas $r = 1$ for class i means that every item from class i is labeled as belonging to class i. Though, precision says nothing about the number of items from class i that are mislabeled (fn) and recall says nothing about the number of items that are incorrectly labeled as belonging to class i (fp).

Both precision and recall are obviously relevant for a multi-label classification problem and, for this reason, the F_1 score appears as being the best evaluation metric for this project. A good retrieval algorithm will maximize precision and recall simultaneously and, consequently, good performance on both is favored over extremely good performance on one and poor performance on the other. Note that all Kagglers participating to this competition use the F_1 score as evaluation metric. Thus, models can directly be compared and participants easily ranked.

II. Analysis

Data Exploration

The various datasets and inputs for this competition are available in the data section of the Kaggle competition webpage[1]. There are 6 different datasets for this competition:

- `train.csv`;
- `train_photo_to_biz_ids.csv`;
- `test_photo_to_biz_ids.csv`;
- `train_photos.tgz`;
- `test_photos.tgz`;
- and `sample_submission.csv`.

There are 9 different business attributes that are encoded as integer ranging from 0 to 8:

0. `good_for_lunch`;
1. `good_for_dinner`;
2. `takes_reservations`;
3. `outdoor_seating`;
4. `restaurant_is_expensive`;
5. `has_alcohol`;
6. `has_table_service`;
7. `ambience_is_classy`;
8. and `good_for_kids`.

The `train.csv` file provides the list of labels for each business id. As mentioned previously, each business has multiple photographs. The correspondence between the photo id and the business id is given in `train_photo_to_biz_ids.csv` (`test_photo_to_biz_ids.csv`) for the training (test) dataset. The photographs for the training (test) dataset are compressed and combined in `train_photos.tgz` (`test_photos.tgz`). All images have `jpg` format and are named after their photo id. There are 234,842 (237,152) photos and 2,000 (10,000) restaurants in the training (test) dataset[2]. Finally, a template file (`sample_submission.csv`) provides the format for the submission. Predictions enclosed in this file are used by Kaggle to calculate the model performance and in turn rank participants.

Table 1 features a small sample of the training dataset. It relates business id, labels and photos id. The number of photos available for each business is calculated and reported in the last column of Table 1. One can see that this statistic varies from one entry to the other. The histogram of the number of photos

	labels	photos id	# photos
business id			
1000	(1, 2, 3, 4, 5, 6, 7)	[438623, 325966, 227692, 407856, 368729, 16319…	54
1001	(0, 1, 6, 8)	[298536, 20346, 8457, 308694, 349310, 407838, …	9
100	(1, 2, 4, 5, 6, 7)	[338465, 328433, 243861, 361777, 127198, 46652…	84
1006	(1, 2, 4, 5, 6)	[46472, 341947, 396253, 75316, 42330, 244095, …	22
1010	(0, 6, 8)	[118251, 219940, 27517, 8578, 148347, 433559, …	11
101	(1, 2, 3, 4, 5, 6)	[13736, 393696, 286907, 86169, 243460, 254663,…	121
1011	(2, 3, 5, 6)	[372371, 116870, 411981, 208597, 127752, 18839…	70
1012	(1, 2, 3, 5, 6)	[287385, 232258, 388225, 151345, 417121, 32754…	37
1014	(1, 2, 4, 5, 6)	[407910, 33911, 269241, 374218, 256236, 296370…	32
1015	(1, 5, 6, 7)	[456770, 44056, 128542, 373344, 87938, 148452,…	145

Table 1: Training dataset. The labels, photo id and number of photos are given for each business id. This table is extracted from the `eda.ipynb` Jupyter notebook.

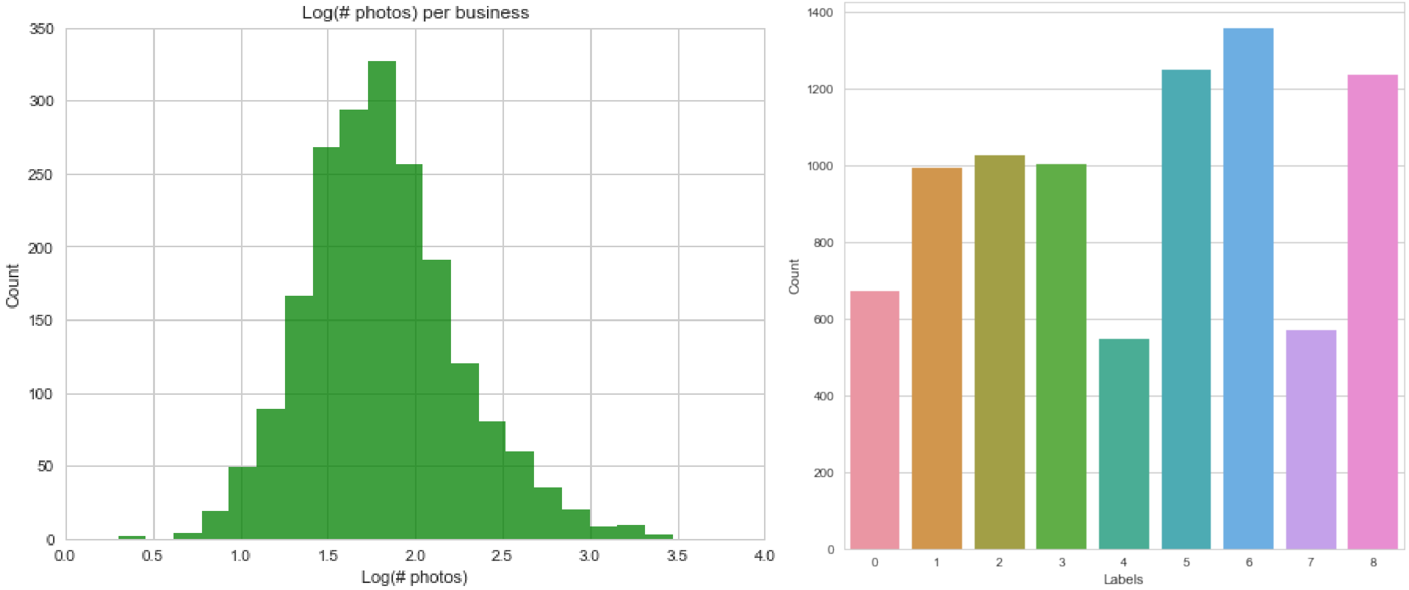

Figure 1: Training dataset statistics. Left: histogram of the number of photos per business. Right: bar plot of the label frequency. These figures are extracted from the `eda.ipynb` Jupyter notebook.

per business for the training dataset is shown in the left panel of Figure 1. Some businesses have very few photos (as low as 2 photos) whereas other have thousands of photos. There is on average 117 photographs per business. The label frequency is shown in the right panel of Figure 1. The least used attributes are `good_for_lunch` (#0), `restaurant_is_expensive` (#4) and `ambience_is_classy` (#7) whereas the most used attributes are `has_alcohol` (#5), `has_table_service` (#6) and `good_for_kids` (#8).

Exploratory Visualization

Figure 2 shows 24 randomly chosen photos that has been tagged as `good_for_kids` (#8). One can see that these photos provide rich local business information. Some photos capture the ambiance/decor of a place whereas other exhibit the food and drinks that are served. Teaching a computer to understand the context of these photos is clearly not an easy task. Objects can easily be misinterpreted and subsequently

[1] Visit `https://www.kaggle.com/c/yelp-restaurant-photo-classification/data`.

[2] The datasets are quite large. Both the training and test tar archive files have a size of about 7 GB.

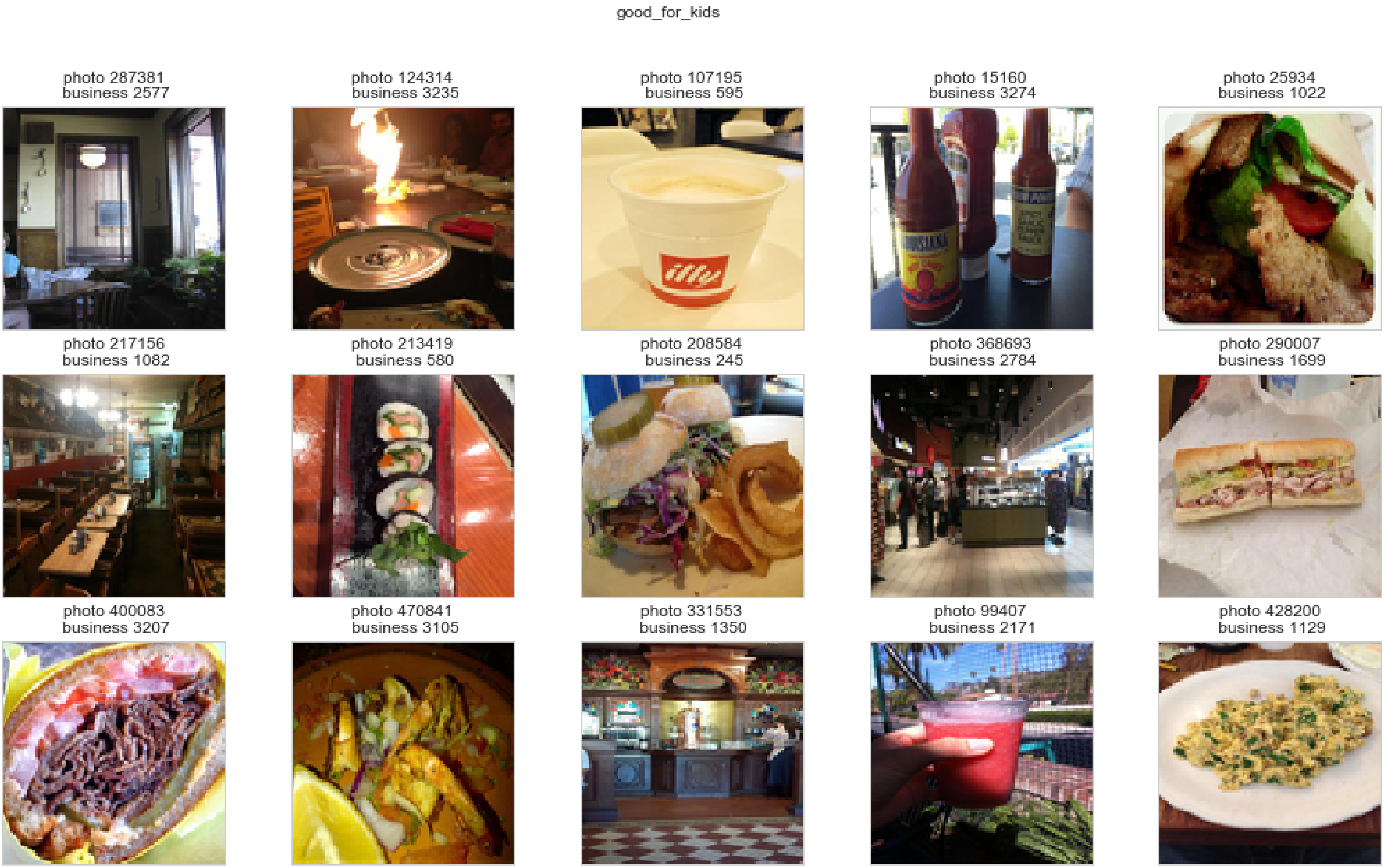

Figure 2: Photos of the training dataset. Each business has been tagged as `good_for_kids` (#8). These images are extracted from the `eda.ipynb` Jupyter notebook.

wrongly classified. For example, condiment bottles that can be seen on photo #15160 (1st row and 3rd column) could easily be interpreted as alcohol bottles by a model. Also, it is worth noting that labels are annotated by Yelp users and hence are subjective. To illustrate, some people think that Japanese cuisine is appropriate for kids while other might not. It follows that an establishment that serves sushis (2nd row and 2nd column), for instance, will likely not be labeled as `good_for_kids` (#8) by every Yelp users who have eaten there and submitted photographs.

Algorithms and Techniques

Neural networks have proven to be incredibly efficient at classifying images and often outperform other machine learning algorithms at this task. It comes then as no surprise that deep learning models are used extensively in this project. One now faces two options: i) build and train a deep neural network from scratch or ii) use transfer learning[3]. The properties of the dataset such as its size and nature usually dictate the type of approach to adopt. The Yelp dataset being both large and complex, it would be unrealistic to train a deep neural network model from scratch given this task would require fine expertise and enormous resources. Also, deep neural networks that have been pre-trained on large and diverse dataset like ImageNet[4] capture universal features in its early layers that are relevant and useful for most computer vision problems. Thus, leveraging such features allows to reach a better accuracy than any method that would rely only on the available data. For those reasons, transfer learning is a better approach for this project

[3] Machine learning technique where a model trained on one task is re-purposed on a second related task.

[4] Large visual database designed for use in visual object recognition software research.

The next step entails the selection of the most relevant pre-trained model for the problem domain. Four state-of-the-art deep learning models whose weights have been pre-trained on the ImageNet database are considered here. For each model, the bottleneck features[5] are extracted and used as inputs of a very simple classifier. Each classifier is independently trained and their performance is evaluated on an unseen set of features. Based on these results, the best pre-trained deep learning model is selected and used as a fixed feature extractor.

Before feeding the bottleneck features to a classifier, one needs to address the multi-instance aspect of this project. There are essentially two options: i) derive a feature vector for each instance and combine them accordingly to get one feature vector per restaurant or ii) assign to each instance the label of its corresponding restaurant, proceed to classification and average the output probabilities for each label. Both scenarios are investigated in this project.

Finally, a classifier is trained and predictions are made. Two models are considered for the classification task: i) a multi-layer neural network with a final layer containing one node for each label and ii) XGBoost (gradient boosted decisiontrees). It is worth noting that a neural network automatically accounts for eventual dependencies among labels because it shares weights for the different label learning tasks. For the other model, label dependencies are handled through classifier chains.

Benchmark

III. Methodology

IV. Results

V. Conclusion

[5] Last activation map before the fully connected layer.