Machine Learning

Social Media Profiling

Daniil Filienko
School of Engineering and Technology
University of Washington
Tacoma, WA
daniilf@uw.edu

Winston Palace
School of Engineering and Technology
University of Washington
Tacoma, WA
wpalace@uw.edu

Justin Goding
School of Engineering and Technology
University of Washington
Tacoma, WA
jgoding@uw.edu

ABSTRACT

This concise ML report presents the results of what our research team learned about various methods that could be applied for predicting age, gender, and psychological characteristics [OCEAN] of a social media user, based on the image, likes data, and social media posts, in addition to various derived sources data, such as LIWC, producing a psychological snapshot based on the user's social updates and other posts available on the social media page. We used CNN for image classification, succeeding in producing above-average results in gender prediction, and various styles of models utilizing Logistic Regression for both text and relational data sources, producing high accuracy results in both the gender and age prediction for the users, utilizing both imported and custom implementation of the general LogRes algorithms.

CCS CONCEPTS

• Machine Learning • Deep Neural Networks • User Profiling

KEYWORDS

Social Media, Data Processing, Deep Learning, CNN

1.INTRODUCTION

In the following paper, we intent on presenting the results of a 10-week course of Machine Learning, where we learned about various methods that could be used to applied for predicting various traits of a user for the purpose of profiling Facebook users based on the profile image, textual updates, and user's likes references. The goal of the study was to produce statistically significant results in predicting the various traits of the user using modern Machine Learning methods with the accuracy exceeding that of the baseline, produced by averaging all available data and predicting test data based on the average values of the corresponding traits. The results were encouraging, as we were able to produce an ensemble utilizing all of the basic data sources, capable of predicting the gender of a user with an accuracy of 81%, age with accuracy of 60%, and performing a regression for OCEAN traits with below-average RMSE metrics. While we did not utilize all of the models presented in the course, we achieved best and most consistent results with CNN in the image classification, LogRes with Text preprocessing for Text, and instance-based learning (KNN) for the user's profile likes relation. Following paper will present the methodology, results, and chosen evaluation metrics in the greater detail, explaining our choice of models per the particular data source.

2.METHODOLOGY

The following section contains methodology followed by the members of our team in order to achieve our results in predicting various traits of a user.

2.1 Image

The first method for gender image classification was a convolutional neural network to determine the gender of an image. The model consisted of 14 layers. 3 layers for randomizing the zoom, rotation, and flipping on the image. The 4th layer was for conversion to grayscale. Then there were several layers of convolution followed by a dropout. The randomization of the images helped to keep the model from overfitting, dropout also contributed to this. Finally, there were two dense layers, with one being the output. It used sparse categorical cross entropy as the loss function and accuracy as the metric. The second method of gender image classification was built upon the first but with the addition of trying to split categories by the number of faces in the image. There were 6 categories, maleNoFace, maleOneFace, maleManyFace, femaleNoFace, femaleOneFace, femaleManyFace. The third method of gender image classification was similar to the first method with the exception of being adjusted to use binary focal cross entropy. This is because the genders given from the data were binary, so categorical cross entropy was not needed. Additionally, less convolutional layers were used, and more dense layers were added to try to utilize deep learning more effectively.

2.2 Relations

The first method used for relations gender classification was K-Nearest Neighbor (KNN), with a K value of 25. The data was organized into python dictionaries with user ID mapping to gender and a dictionary of user ID mapping to a dictionary of post IDs liked by the user. A distance formula of $d = n + m - 2s$ was

used to calculate the nearest neighbors, where d is distance, n is the number of liked posts by the user whose gender is to be predicted, m is the number of liked posts by the user for which distance is being calculated, and s is the number of posts in the intersection set of the two users' liked posts. This formula effectively calculates the Euclidean distance between two points in binary space given sparse vector positions for the points. Once the 25 nearest neighbors had been determined the gender prediction was a result of simply taking the mode of the 25 neighbors. Few users had liked the same posts so this resulted in most users' nearest neighbors just being those that had liked the fewest posts and therefore took them the least distance from the origin point causing most predictions to simply be the mode of the 25 users with the fewest likes. To solve this issue a weight was given to s in the distance formula to give a higher value to occurrences of two users liking the same posts, a value of 15 was given to this weight so the formula then became d = n + m - 2(15s). Since this worked well it meant that similarly liked posts mattered when determining the closest neighbor so the previous distance formula was swapped out for the Jaccard distance to more effectively calculate distance based on similarities. The prediction was then changed to be the weighted average of the k nearest neighbors where the weight of a given neighbor was 1 / distance, for gender classification this average was rounded to 1 for female or 0 for male. Gender classification's k value was changed to 31 which saw a slight improvement over the previous value of 25. This method was also then adapted to predict age where a k value of 25 was again used and the prediction was the weighted average of the 25 nearest neighbors. This was also used to attempt to predict the OCEAN scores of users with a k value of 17 and then later a k value of 25, but both of these only achieved baseline accuracy. Next, the approach to these predictions began from scratch with an entirely new model. A logistic regression model using a sigmoid function was made for the problem of gender classification. Data was once again stored in python dictionaries with one dictionary mapping user IDs to gender and another mapping user IDs to a dictionary of liked post IDs. Nodes for the model were represented as a dictionary of post IDs mapped to the edge weight for that input. A learning rate of 0.3 was used, edge weights were updated after every example instance using the formula: wi=wi-1+td-odxi,d, where w is the edge weight, is the learning rate, td is the expected output, od is the prediction, and xi,d is the input value. The model was trained over the data set once and saved to a text file, this model worked well right out of the gate. This logistic regression function was adapted to predict OCEAN values by simply multiplying the prediction by 5. This method for OCEAN prediction was then replaced with a linear activation function, then a rectified linear unit (ReLu) function. With the ReLu function, learning rates of 0.1, 0.3, 0.5, and 1.0 were tested and for one attempt the model was trained over the data set 4 times with a learning rate of 0.3 but none of these attempts were successful. The logistic regression program was then added onto to work as a softmax regression for the task of predicting age in the given age buckets where each output node was one of the age buckets. A sigmoid function was once again used to get a value for each output node, with a learning rate 0.3, updating weights after every training example, and training over the data once. The output values were run through a softmax regression activation function to normalize the outputs, and the node with the highest value was chosen for the prediction. This model also worked quite well right away.

2.3 Text

2.3.*1 Logistic regression with Text Preprocessing over TF-IDF tokenized text.* As it sounds, that method allows to perform Logistic Regression on the tokens acquired through performing TF-IDF tokenization on the textual data, which improves over traditional Bag of Words approach by performing a form of normalization of the frequency of words in a provided document, rather than just reporting the count of certain words in the given preprocessed text, as BoW does. Preprocessing the text by low casing the words and not removing stop words happened to produce highest for me overall accuracy than stemming or lemming the words for both age and gender.

2.3.*2 BERT Preprocessing with Keras Deep Neural Network. BERT.* Following an online tutorial, I also implemented multilayer NN with first layer accomplishing BERT-specific text preprocessing to put it in certain proper format and the second layer converting the text token to BERT vector (a word embedding model trained by Google on Wikipedia and books to encapsulate the "meaning" of the word in a numerical 768-dimensional vector). I chose BERT, because differently from simple CountVectorizer or TD-IDF, encapsulating the probability of word of occurring, BERT encoding would be capable of encoding "contextual relations between words (or sub-words) in a text" within a vector of 768 continuous numbers (Horev, 2018). In order to use BERT, I utilized tenrflow_text library to make a call to the BERT preprocessing and encoding models trained specifically to uniformly preprocess and then turn text words into 768-dimensional vectors called word embeddings, with each value within a vector describing the measure in which certain feature, such as how "country-like" or "person-like" is the word. There are multiple possible models, and I utilized the 'simplest' one. According to Dhami (2020), in a BERT token, "original word [can be] split into smaller subwords and characters. This is because Bert Vocabulary is fixed with a size of ~30K tokens. Words that are not part of vocabulary are represented as subwords and characters." In order to produce a contextually dependent prediction, rather than just operating on the individual embeddings, encapsulating the 'meaning' of the token, "a simple approach is to average the second to last hidden layer of each token producing a single 768 length vector" within the imported tensorflow_text BERT encoding model. Therefore, that allowed to encapsulate the complex relations between the words that provides additional data for the model, potentially allowing to acquire higher accuracy while requiring more complex ANN model.

2.3.3 Doc2Vec with Deep NN or Logres. Doc2Vec failed with both a 2-layer Keras NN and logres, performing poorly due to the tendency to overfit of the Doc2Vec, which I could not transfer and

attempted to train over the new document, hoping for similar encoding to the ones on which original models were trained on, however it led to lowew accuracy. I think the reasons was poorly organized Neural Networks with a varying doc2vec model, trained on limited training data, due to the data-intensity associated with linear gradient, utilized to train weights of a NN to produce better result, more closely fitting provided data.

3.Dataset and Metrics

3.1 Image

The image dataset for the first method was broken up into two subfolders 0 and 1 to represent men and women for the categories. Keras would then take those categories and split them up further to create training and validation data. Results were judged based on the accuracy of the prediction. The second method broke up the image dataset into 6 categories that have been previously mentioned. Keras would split these up into training and validation datasets as well. They were also judged by accuracy. The third method used a setup that was the same as the first.

3.2 Relations

The relations dataset came in a csv with two columns, the first column was user ID and the second column was the ID of the liked post. This csv was parsed and organized into a python dictionary where user ID mapped to a python dictionary of liked post IDs. This allowed for iterating over one user's liked posts and quickly checking if that same post appeared in another user's likes. The profiles were organized into another dictionary of profile IDs mapped to whichever attribute was being predicted. Then for testing the set of profile ID keys were placed into a python list and shuffled, then 10% of the keys were saved for testing while the other 90% was used for training. If a good accuracy was achieved with this internal testing, then the model was trained over the entire data set to be tested against the real test data.

3.3 Text

Text training dataset was consisting of 9500 text instances, LIWC evaluations, relation vectors, and images, associated with 9500 users, a small public test dataset, and a 1000+ secret testing sample, to produce accuracy evaluations of the various student models. Text dataset consisted of all the text-based social media posts of the user combined in one line without data of where one post ends and another begins. LIWC dataset was consisting of certain psychological characteristics traditionally associated with the text patterns and vocabulary utilized by the user, utilizing an algorithm that, according to Chung *et al* (2012) "can be considered to be a tool for applied natural language processing since, beyond classification, the relative uses of various LIWC categories can reflect the underlying psychology of demographic characteristics, honesty, health, status, relationship quality, group dynamics, or social context." I combined the datasets with the values of predicted features, associated with the given IDs, and utilized that dataset to train an NLP model.

4.Results

4.1 Image

The first method for gender image classification had the greatest success. Its training accuracy was around 70% and was deployed on the VM with a 68% that beat the baseline on the scoreboard. The second method for gender image classification was a lot more successful with training accuracy but testing after training showed that it was not good at determining gender but rather how many faces were in an image. The third method for gender image classification got nearly identical results as the first method but was never used on the scoreboard; however, it was easier to work with and saw continued work after the results were deemed the same.

4.2 Relations

The first relations method attempt of KNN with no similarity weighting performed quite poorly on gender classification with an accuracy around 55%, but once the similarity weight was added it improved to an accuracy of 72%. Once it was updated to use Jaccard distance, take the weighted average of the nearest neighbors, and given a k-value of 31 the accuracy went up to 76%. This same method but with a k-value of 25 was used on age prediction with a poor accuracy of around 55%. It was also used to predict OCEAN values, one attempt with a k-value of 17 and another with 25, both only achieved baseline accuracy. The logistic regression model for gender performed internally quite well right away with an accuracy of 76%, being short on time and previous internal tests being closely reflective of what to expect on the actual test data, this method was added to the ensemble gender prediction without first being tested alone against the test data. The ensemble accuracy was tested again after the change and accuracy was the same before and after which is expected since the logistic regression had the same gender prediction accuracy as the KNN. The logistic regression was used on OCEAN values performing best when simply multiplying the output of the sigmoid function by 5 to match the 0-5 scale of the OCEAN values, this achieved an RMSE of 1.5. The linear unit which was expected to be more appropriate for this task performed worse with an RMSE around 10, the ReLu function achieved an RMSE of 9.4, the different learning rates and the attempt with 4 epochs did not make much difference, all scoring around the same accuracy. The softmax regression for the problem of age prediction also worked decently well right away with an accuracy of 63%.

4.3 Text

Source	Age (Accuracy)	Gender (Accuracy)
Baseline	0.59	0.59

Text	0.61	0.75

Figure 3: Results for the text

4.3.1 Logistic regression with Text Preprocessing over TF-IDF tokenized text. I think that method succeeded because of its simplicity, and was better without stemming potentially because stemming removed certain patterns that happened to involve a way certain word is used in a particular type of users, because stemming attempts to simplify the word to the words 'meaning', potentially removing certain relationship between the words that maybe could be noticed in the tendency of certain forms of words to be used together by a group of users. TF-IDF performed better than BoW because the former provides additional data about a relative use of the word, allowing Logres to acquire additional data about the probability of a word to occur in the document in particular form.

4.3.2 BERT Preprocessing with Keras Deep Neural Network. BERT performed encoding attempting to depict the meaning of the words, rather than the frequency, differently from TF-IDF, being trained on a large dataset of books and Wikipedia articles by Google, attempting in its encoding to encapsulate the words 'similarity' to related concepts. However, it was hard to deploy on VM, as it required to communicate with an external TensorFlow service to preprocess the text and convert the text into appropriate embeddings, on which the NN model was trained, but BERT's uniformity and complexity meant that encoder could provide enough data per word to permit a large neural network to potentially produce gender prediction with a relatively high accuracy. However, because of my lack of practice at hyperparameters tuning and relatively small dataset, I achieved a limited accuracy of 69% on the training dataset. I suppose that in addition to the lack of skill, small dataset could be a problem, as in order to recognize a complex class boundaries, associated with NLP classification in general, due to its complexity, much weights adjusting is needed in order to reach local maxima of highest accuracy, which meant that more complex model would be required with additional layers, however in order to prevent overfitting, the batch size would be required to be small and the amount of layers should not exceed the threshold where the model starts to memorizing the data, rather than generalizing the classification boundaries present in the dataset provided to the model.

4.3.3 Doc2Vec with Deep NN or Logres. Doc2Vec failed with a 2 layer Keras NN and logres, accuracy of 13 and 29 percent for gender and accuracy, due to overfitting and learning certain trends discovered by doc2vec in the training dataset, but not present in the testing dataset. Also, when we trained doc2vec on the fly on the testing data set, rather than storing and loading the doc2vec model utilized for training the NN model, the accuracy fell with NN because of different doc2vec encodings, explaining why BERT had higher accuracy, since it had uniform way of extracting word embeddings, eliminating additional variation between the test and training doc2vec formats, which could vary in the failed setup, since they could attempt to pick up different features of train/test data throughout the multiple runs, because of random weights and tendency to overfit, associated with the poorly organized NN models, of which doc2vec was an example, which were trained on limited training data.

5.Conclusion

5.1 Image

In conclusion, the baseline for gender was able to be beaten with convolutional neural networks and the image dataset. The best method involving the image dataset was to use a convolutional neural network with sparse categorical cross entropy as the loss function with binary focal cross entropy being a close second.

5.2 Relations

For relations, KNN proved to be a simple and quick way to beat baseline accuracy for gender, but failing for age, and only achieving baseline for OCEAN. While KNN did well right out the gate it seemed to offer few opportunities for optimization and higher accuracy, so the model was abandoned in favor of logistic regression for gender, and softmax regression for age, both doing quite well right away. These were promising models that most likely could have benefited from some more time to optimize them. A multilayer perceptron model may be a way to achieve even better accuracy but would require much more time to train and tweak.

5.3 Text

5.3.*1*: LIWC *with Random Forests.* Ensemble's high accuracy and good performance with relatively small dataset led me to think about performing a gender prediction with random forest over the LIWC features, which, while being continuous, I think would not be linearly separable, preventing me from utilizing Random Forest model ensemble. I presume that it may not be linearly separable due to low accuracy of linear SVMs over the dataset, which would have been higher if it was linearly separable, but produced gender predictions with a low accuracy that is slightly above the standard of 59% percent.

5.3.*2*: *GPT embeddings with NN.* Popularity of chatGPT, utilizing GPT word embeddings, and transformers in general, may be a good reason to attempt to utilize a GPT embedding or another alternative to BERT that could be trained on a dataset corresponding better to a social media type of communication medium, because BERT's training on books and Wikipedia, which primarily utilize more formal language, could potentially prevent it from producing accurate gender prediction, because it could associate a word with a concept different from the one that a word may be encompass in non-formal settings of the social media context.

REFERENCES

[1] Chung, Cindy & Pennebaker, James. (2012). Linguistic Inquiry and Word Count (LIWC): pronounced "Luke",… and other useful facts.. 10.4018/978-1-60960-741-8.ch012

[2] Horev, R. (2018, November 17). Bert explained: State of the art language model for NLP. Medium. Retrieved March 10, 2023, from https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270