Is it possible to recognize Apple employees by their LinkedIn profile picture?

Thanakij Wanavit[1][a] and Leslie Klieb[1][b]

[1]*Business and Technology, Webster University Thailand, 1 Empire Tower, Sathorn Rd, Bangkok 10120, Thailand*
nwanavit@hatari.cc, kliebl@webster.ac.th

Keywords: Convolutional Neural Networks, Emotion Recognition, Employment at Apple, EmoPy, Random Forest Classifier, Classifying Employment from Emotions.

Abstract: Samples of images from the portraits on the profiles of members of the social media site LinkedIn who live in the Bay Area of St. Francisco were collected and analyzed by the EmoPy package for the presence of seven emotions. A Random Forest classifier used these probabilities to predict if the members were employed by Apple or not. Accuracy reached around 62% compared with a naive error rate of 50%. An error analysis shows that this result is significant and robust.

1 INTRODUCTION

Is it possible to find out from profile pictures on a social network where people work? This position paper will suggest that the use of neural networks and Machine Learning can indeed accomplish this in some circumstances. Employment at one of the FAANG (Facebook, Apple, Amazon, Netflix, Google) companies as a software developer is surrounded by an aura of mystique. The Internet abounds with people, especially computer science students, searching for answers to the question how to get hired by one of those prestigious companies. Most answers on the Internet focus on skill sets, ability to answer interview questions, and other cognitive advantages. The Research Question of this work tries to answer a related question focusing on non-cognitive differences. Is it possible to find differences between male software engineers working for Apple, Inc. and those working for other companies in the San Francisco Bay Area that are sufficient to classify those employees as Apple or non-Apple employees, and that have no connection with background, skills, or other obvious cognitive characteristics? In order to answer that question, a sample of LinkedIn profiles was scraped from the LinkedIn website searching for current employment at Apple or not Apple, and for location in the Bay Area. No other information was collected or retained. The research uses Machine Learning (ML) and neural network technologies to determine a numerical estimate of the emotional content present in the photographs. Note that those images are photographs chosen by the employees themselves with the purpose to present a professional look that is, in combination with the other content in the profile, attractive for headhunters and others interesting in hiring developers. The Random Forest statistical algorithm is trained on a set of those images. It is found that the software is indeed able to make classifications with a small but significant success rate.

2 WHY APPLE?

An experiment like this has only a chance to succeed when a few conditions are met. The core of employees must be in some sense sufficiently distinct from the employees at other companies. This is in the Bay Area true for Apple.

First, Apple is sufficiently large that a reasonably size random sample could be collected. That is not true for smaller companies.

Secondly, the company is focused on innovation. It renews its products offerings every one or two years. Steve Jobs established for this purpose a culture of competition between groups inside the company. This is a large difference with other companies with more established products. It will attract a special kind of developer.

Thirdly, working for the Apple company is glam-

[a] https://orcid.org/0000-0001-7291-394X
[b] https://orcid.org/0000-0002-0881-5330

orous and they can pick their staff to a much higher degree than smaller software companies can do. Differently from what might be thought, Apple does not dominate the software developing industry in Silicon Valley. There are less developers working at Apple than at Google in the Bay Area. It was estimated that in May 2016 there were 41,490 application software developers, 28,670 systems software developers in the Bay Area. An estimate for the number of software developers at Apple is around 12000-16000. Therefore it is possible for Apple to be picky in who they hire, not only in skills but also as fit to a quite different company culture. (Anonymous, 2017)

3 PROCESS OVERVIEW

In this short section an overview of the analysis process is given. Details and references are in the Methodology Section. The steps are the following:

- Collect LinkedIn profile pictures
- Three samples:
  - Male Apple employees in the Bay Area of San Francisco with occupation computer software (Apple=true, Control=false)
  - Male non-Apple employees in the Bay Area of San Francisco, same occupation (Apple =false, Control=false)
  - Male employees, any occupation, from the USA (Apple=false, Control = true)

  Those three categories are shown in Table 1.
- The EmoPy package provides in one package object detection and recognition of seven emotions: anger, fear, calm, sadness, happiness, surprise, and disgust
- Statistical analysis of the differences between the three samples

4 LITERATURE OVERVIEW

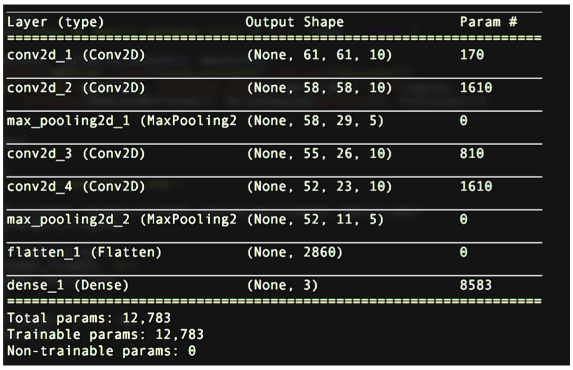
Every company depends for its survival on a steady stream of new personnel that via a process of onboarding will become a good fit in skills, character, motivation, assimilation of the organizational culture, and other traits. It is therefore to be expected that there are differences between employees who work for a company like Apple and for other companies. The selection will start already at the reading of the resumes and looking at LinkedIn profiles, and can lead to differences in the two groups that are amplified at every stage, from interview and hiring decision, to turnover and retention (Marsden, 1994) Implicit or explicit biases might be present that influence this process (Bendick and Nunes, 2012).

Social media profiles have been found to represent the personality of individuals (Back et al., 2010). Relationships between characteristics of pictures and personality were found in Facebook profile pictures by (Celli et al., 2014). As different companies can attract different type of people and those differences can be discernible by AI in the profile pictures on Social Media, it is an interesting issue to see if in the reverse direction, the profile pictures are sufficiently distinct so that it is possible to classify the employees according to the company.

5 METHODOLOGY

Sampling was automated using Selenium. It was carried out by searching LinkedIn from an apparent location in the USA at the ”search for people” tab with filter ”company: Apple, location: San Francisco Bay Area, occupation: computer software, sex: male”, and keeping every second result. Similarly, results were collected for non-Apple employees in the Bay Area with the same filter but not working for Apple. A control group had as filter ”is in the United States” and not working for Apple. The search routine stored the profile picture, occupation, and company but no other information. There were 851 Apple employees, 416 non-Apple employees, and 517 ”control” LinkedIn members in the sample.

The freely available open source EmoPy package (Perez, 2017) from Thoughtworks (Perez, 2018b), (Perez, 2018a) was the main tool to extract emotions from the LinkedIn image set. Extraction of the deeper information in an image is possible using a combination of a trained object detection Convolutional Neural Network (CNN) followed by classification CNN. This is demonstrated by an emotion extraction package such as EmoPy (Perez, 2018c). EmoPy uses machine learning to recognize the presence and strength of the expression of seven universal (not depending on cultural artifacts) standard emotions (anger, fear, calm, sadness, happiness, surprise, and disgust) in every image. The tool itself consists of a number of convolution neural network layers interspersed with Pooling layers that decrease the amount of information when it gets too big too handle. The EmoPy package comes with a series of default convolutional pretrained models. One of the pre-trained models available in EmoPy is the Facial Expression Recognition (FER) model which is a convolutional network trained on the FER dataset (Barsoum et al., 2016), based on the FER2013 image set (Perez, nd). A slightly modified new model was made for this work in order to make it possible that the model predicts seven outputs instead of four, by stacking together two models that were published at the EmoPy website. This was not available at the time of this work, but a new upload to the EmoPy packages seems to make this currently possible.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 61, 61, 10)        170
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 58, 58, 10)        1610
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 58, 29, 5)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 55, 26, 10)        810
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 52, 23, 10)        1610
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 52, 11, 5)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2860)              0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 8583
=================================================================
Total params: 12,783
Trainable params: 12,783
Non-trainable params: 0
```

Figure 1: Parameters EmoPy

Before emotions can be recognized, the system must first separate the face itself from the surrounding background. A CNN used by Emopy uses the same concepts as were proposed in the well-known YOLO Paper((Redmon et al., 2016)) to recognize objects with four methods simultaneously, namely pose-robust feature extraction, multi-view subspace learning, face synthesis based on 2D methods, and face synthesis based on 3D methods, all in one fast pass. A convolutional neural network (CNN)like EmoPy can do these simultaneously together with the emotion recognition in YOLO mode (combining the face recognition itself and the emotion recognition) in one pass while using the Graphics Processor Unit for speed. The Python code ran in a Jupyter notebook in a virtual machine provided by Google Colab (Colaboratary, nd). The data inside Colab is automatically backed up on Google Drive and is available publicly.

Each of the seven emotional features in each LinkedIn picture is assessed by the CNN and gets a score which indicates the strength of that emotion. These scores are normalized so that they sum to 100. This happens at the end of the pipeline in a flattening layer. Collectively, the strength of those emotional features should summarize all the useful information taken into account from the image in this research while reducing the amount of data fed into the final layer of image classifier. The last layer is a fully connected layer (Perez, 2018a), and converts the strength of the found emotions into probabilities, with their sum per image equal to 100% via a SoftMax activation function.

$$softmax(z_i) = \frac{exp(z_i)}{\sum_{j=1}^{k} exp(z_j)}, \qquad (1)$$

where k is the number of classes, 7 in this case, i=1...k, and weights and bias have been neglected because they are difficult to assess here. The SoftMax function is a generalization into more dimensions of the sigmoid function from logistic regression that makes sure that the total probability is smaller than 1. z_i is the strength of emotion i.

Also, the Keras-Tensorflow library of low-level and high-level API routines (Abadi et al., 2015) was used in this research. Tensorflow also uses the GPU and Colab.

The research photos and temporary data logs of this work were stored in the S3 object store, a service from Amazon Web Service which allows user to store and distribute files stored into folder-like buckets. S3 has multiple advanced feature which are beneficial to this type of research including data redundancy, version control, accelerated data transfer, high-speed web hosting, and permission control. The DynamoDB database service from Amazon was used to store all data, also the non-structured data. The service is low cost and the performance is sufficient for real time data processing used by the Python code in this research. The database is highly redundant with safety measures built-in to prevent accidental data losses. All forecasts data from EmoPy and other binary algorithms are stored in DynamoDB within S3 (Amazon, n d), (Sivasubramanian, 2012)

6 ANALYSIS

The output of EmoPy is passed in its last layer through the SoftMax activation function in in order to convert the scores of the emotion feature set into a probability. The input for the binary classifier is the output of the EmoPy neural network which consist of the probability of each of the seven emotions in each image. The output of the binary classifier is the predicted probability of the person working for Apple or not. This is a number between 0 and 1. 0 means the person is predicted as certainly not working for Apple and 1 means the person is predicted with certainty to work for Apple. If the output 0.5 or higher, the output is a prediction that the person is working for Apple, just like in binary logistic regression methods. If the result is correct, then the algorithm gets 1 point for making the correct prediction and if incorrect, zero points.

Table 1 gives the mean and standard deviation of the probabilities for each emotion for all samples. It is seen that only ”calm” has a narrow peak and that disgust and sadness have a very wide shape. Differences between Apple and non-Apple employees are minimal. This makes classifying difficult.

The quality of a forecast is usually given by the confusion matrix. This is described in Appendix A. Binary logistic regression with SPSS gave a minimal improvement over the naive estimate. A number of other analysis methods were applied, but not reported here because their performance was less than the best performing algorithm, Random Forest, (Breiman, 2001). This is a meta estimator that is fitting some decision tree classifiers on various subsamples of the dataset. It uses averaging to improve the accuracy of the prediction and for control of overfitting.

The Random Forest model was implemented using source code from scikit-learn (Pedregosa et al., 2011) available via (Anonymous, ndb).

The analysis was always done by comparing two sets of data only, Apple versus non-Apple or Apple versus Control, and not by attempting to classify all three sets of data simultaneously. Because we were in charge of the data collection and the choice was made to do only binary classifications, it was advantageous to avoid the accuracy-paradox and balance the samples so that the base accuracy always was 50%. This balancing is not always strictly required by all used algorithms but is recommended (Anonymous, nda).
The sample set was kept balanced by randomly removing some Apple cases from the 851 collected images for Apple so that the total number of cases fed to the model is 416 for the Apple, non-Apple and Control sets. The group of in total 2 * 416 = 832 Apple and non-Apple employees was divided randomly into 666 training cases and 166 test cases (a ratio of 80% versus 20%). This training set of 666 cases was kept at 50% Apple and 50% non-Apple, so that the test set was also balanced. An estimate of the error in the forecast was made by repeating this procedure 1000 times. For each of the 1000 runs, 416 Apple images were randomly chosen from the whole group of 851 collected Apple images and then divided randomly into 666 balanced training and 166 balanced test cases. The model is then trained and made to make another forecast. Results are plotted and differences analyzed using the z-test to see whether the prediction is significantly higher than the 50% baseline that results from equal-size sample sets and its improvement over baseline is a reliable estimate.

Figures 2 and 3 show a scatter diagram of the binary relationships between each pair of emotions for Control=false. The Y-axis is in both diagrams (from top to bottom) surprise, calm, happiness, sadness, disgust, anger and fear. The X-axis for 2 is surprise, calm, happiness and sadness, and for 3 the remaining ones disgust, anger and fear. The orange colors are for Apple employees and the blue colors for non-Apple employees who are also computer engineers

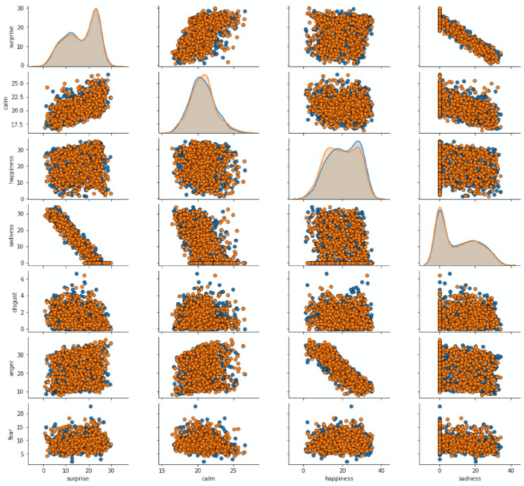


Figure 2: Correlation of the seven emotions for Apple and non-Apple

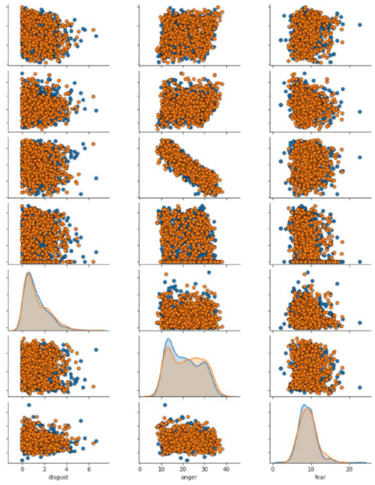


Figure 3: Correlation of the seven emotions for Apple and non-Apple

Visual inspection of binary relationships between all emotions does not show any obviously different pattern for Apple and non-Apple Control employees. A negative linear relationship is visible in the plots between sadness and surprise and between happiness and anger for both groups. The latter relationship is intuitively clear, but there is no clear explanation for

Table 1: Mean and Standard Deviations Probabilities for all samples

			surprise	calm	happiness	sadness	disgust	anger	fear
Apple	Control								
False	False	Mean	17.08	20.72	21.23	10.37	1.28	20.29	9.03
		SD	6.71	1.67	7.55	10.38	1.00	6.94	2.20
False	True	Mean	16.67	20.61	20.56	10.98	1.24	20.76	9.19
		SD	6.83	1.69	7.80	10.44	1.02	7.26	2.31
True	False	Mean	17.01	20.73	19.83	10.25	1.29	21.58	9.32
		SD	6.86	1.73	7.80	10.33	1.05	7.50	2.36

the first relationship.

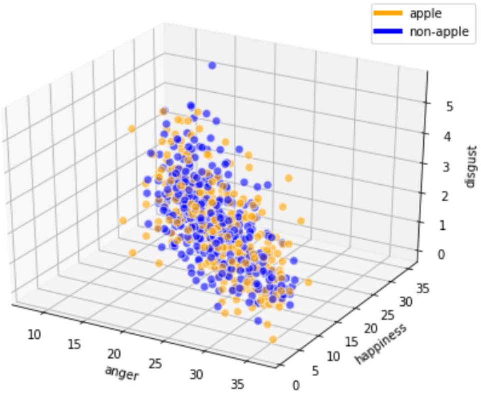


Figure 4: 3D scatter plot for anger, happiness and disgust, for Apple and non-Apple employees

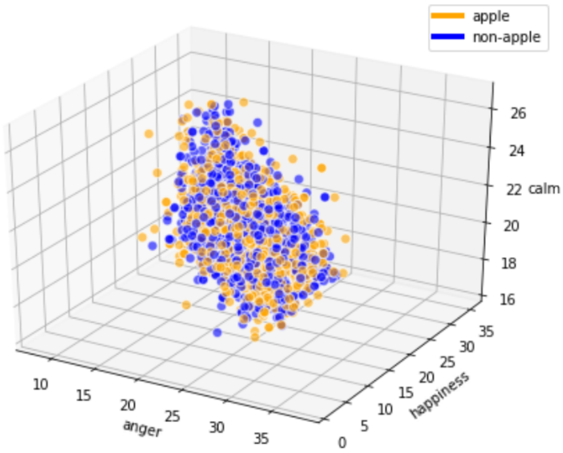
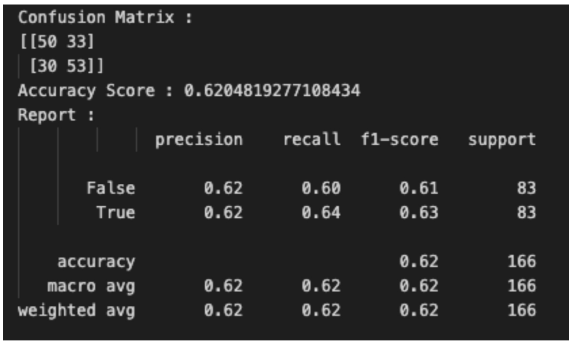


Figure 5: 3D scatter plot for anger, happiness and calm, for Apple and non-Apple employees

Figures 4 and 5 show 3D plots of Apple and non-Apple employees. There is a weak suggestion in these plots that Apple hires more people who score outside the average. The edges of the 3-dimensional clouds contain mostly Apple employees. One could speculate that Apple, as a company that depends on innovation and therefore creativity for its success, is willing to employ people that are less conformist. This would form an interesting future research topic.

One of the 1000 runs yielded, as an example, the following confusion matrix in Figure 6. This is a more or less random, average, example. Its accuracy is very similar to the most frequent score, as can be seem in Figure 7. There is no way to find the most accurate prediction without trying them all out and getting an idea of the distribution of the accuracy scores.



```
Confusion Matrix :
[[50 33]
 [30 53]]
Accuracy Score : 0.6204819277108434
Report :
              precision    recall  f1-score   support

       False       0.62      0.60      0.61        83
        True       0.62      0.64      0.63        83

    accuracy                           0.62       166
   macro avg       0.62      0.62      0.62       166
weighted avg       0.62      0.62      0.62       166
```

Figure 6: Confusion Matrix Apple versus non-Apple

Out of 166 balanced testing cases, 83 from Apple and 83 from non-Apple software engineers, Random Forest gave a forecast of 50 true positives, 53 true negatives, 30 false negative, and 33 false positives. For the thousand runs, the lowest accuracy was 51.03%, the highest 65.4%, the 25th percentile was at 57.48%, the median at 59.24%, and the 75th percentile at 61.00%. The average was 59.18% with a standard deviation of 2.46% and a standard error of $\sigma/\sqrt{n}$, using a sample size of 1000, of 0.078. The z-score of the mean compared with the base rate of 50% is then incredibly large and the null hypothesis that the Random Forest method has a 50% base rate impossible for any reasonable level of significance. The histogram in Figure 7 shows the Gaussian shape

The various emotions contributed in the following way to the classification as in Figure 8.

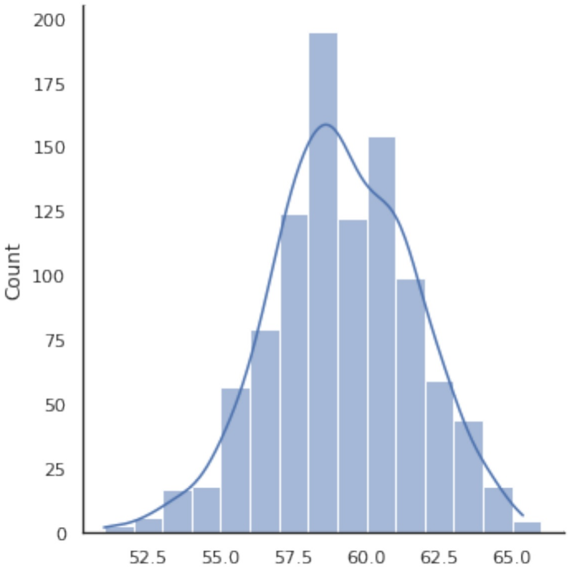


Figure 7: Accuracy in 1000 runs with Random Forest model

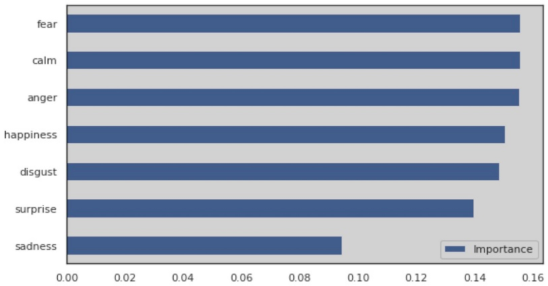


Figure 8: Contribution of each emotion to the Random Forest model

As another metric of the quality of the forecast, the ROC and AUC are shown for the same run as in the Confusion Matrix in Figure 9

A comparison of Apple-employees with the control group gives similar results. There is not much difference between the Control group and the non-Apple group. This provides an argument not to handle this as a classification problem between all three groups.

7 DISCUSSION AND CONCLUSIONS

Only sadness is of less importance in the classification of the Random Forest model, further all emotions seem to play a role, see Figure 8. It is tempting to speculate about the connection of the emotions in the images and the personal characteristics that can make a difference in hiring and retention at Apple. Given that fear, calm and anger seem to make the largest contribution, it is tempting to wildly speculate about a connection with Apple's internal competition-driven organizational culture.

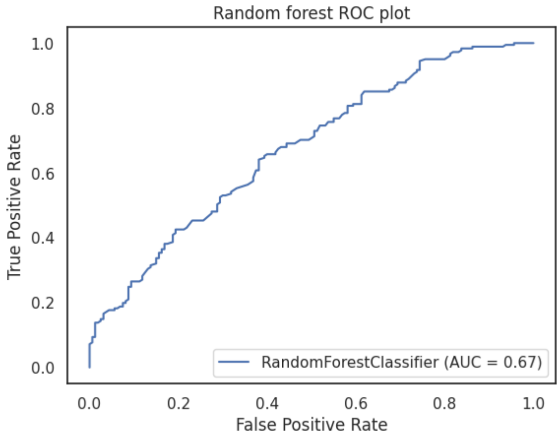


Figure 9: ROC and AUC for one of the runs

If this is true or not, this work has shown that there is a definite connection between the expressiveness of emotions in the LinkedIn profile pictures and employment with Apple or others. However, this exploration is made complicated because there is no simple linear relationship between the probability of the emotions as uncorrelated independents and the probability to work at Apple. If there would have been, then binary logistic regression analysis would have worked, but it did not. We don't know if there are higher-dimensional "islands" of emotions with certain probabilities that are connected with a higher or lower chance to work at Apple, or if the reasons are even deeper. It is, however, plausible that the possible "islands" reflect psychological constructs or biases like age or race more than a bias towards people showing certain emotions. The research shuffled test and training data thousand times and the findings were robust.

The research focused only on the emotions and did not collect any other data about the people on the photographs.In future work we hope to explore if the enhanced labeling of the FER+ image dataset can improve the classification.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/. Software available from tensorflow.org.

Amazon (n. d.). Amazon s3. https://amazon.com/s3/. Information Page.

Anonymous (August 19, 2017). How many software engineers are currently employed in the silicon valley bay area? https://askwonder.com/research/software-engineers-currently-employed-silicon-valley-bay-area-ran93oji9.

Anonymous (n.da). knn and unbalanced classes. https://stats.stackexchange.com/questions/341/knn-and-unbalanced-classes.

Anonymous (n.db). Random Forest Classifiers. https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble.RandomForestClassifier.html.

Back, M. D., Stopfer, J. M., Vazire, S., Gaddis, S., Schmukle, S., Egloff, B., and Gosling, S. (2010). Facebook Profiles Reflect Actual Personality, Not Self-Idealization. *Psychological Science*, 21(3):372–374.

Barsoum, E., Zhang, C., Ferrer, C. C., and Zhang, Z. (October 2016). Training deep networks for facial expression recognition with crowd-sourced label distribution. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, page 279–283.

Bendick, M. and Nunes, A. P. (2012). Developing the Research Basis for Controlling Bias in Hiring. *Journal of Social Issues*, 68(2):238–262.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(3):5–32.

Celli, F., Bruni, E., and Lepri, B. (2014). Automatic Personality and Interaction Style Recognition from Facebook Profile Pictures. In *MM '14: Proceedings of the 22nd ACM international conference on Multimedia*, page 1101–1104.

Colaboratary (n.d.). Frequently asked questions. https://research.google.com/colaboratory/faq.html. Google Colaboratory.

Markham, K. (March 25, 2014). Simple guide to confusion matrix terminology. https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/. Terminology.

Marsden, P. (1994). The Hiring Process: Recruitment Methods. *American Behavioral Scientist*, 37(7):979–991.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Perez, A. (2017). Emopy (0.0.5). https://github.com/thoughtworksarts/EmoPy. Package.

Perez, A. (2018a). Emopy: a machine learning toolkit for emotional expression. https://www.thoughtworks.com/insights/articles /recognizing-human-facial-expressions-machine-learning. Thoughtworks Arts [blog].

Perez, A. (2018b). Recognizing human facial expressions with machine learning. https://www.thoughtworks.com/insights/articles/ recognizing-human-facial-expressions-machine-learning. [blog].

Perez, A. (n.d.). Fer2013 image data set for emopy. https://github.com/thoughtworksarts/EmoPy/tree /master/EmoPy/models. Package.

Redmon, J., Divvala, S., Girshic, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pages 779–788.

Sivasubramanian, S. (May 2012). Amazon dynamodb: a seamlessly scalable non-relational database service. In *SIGMOD '12: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, page 729–730.

APPENDIX

In this appendix an overview is given about terminology related to the confusion matrix. The discussion follows closely (Markham, 2014) A confusion matrix describes the performance of a classification model (or ”classifier”) on a set of test data for which the true values are known.

Table 2: Example of a Confusion Matrix.

N=165	Predicted No	Predicted Yes	
Actual No	No, TN=50	Yes, FP=10	60
Actual Yes	No, FN=5	Yes, TP=100	105
	55	110	165

In this example there are two possible predicted classes: ”yes” (positive) and ”no”(negative). The classifier predicted ”yes” 110 times, and ”no” ”55”. In reality, as given in the ”actual” row, there were 105 cases were ”yes” and 60 cases ”no”

Basic terms are (whole numbers, not rates)

- True Positives (TP): Predicted ”yes”; are ”yes”.
- True Negatives (TN): Predicted ”no”; are ”no”.
- False Positives (FP): Predicted ”yes”; are ”no”. (A ”Type I error.”)
- False Negatives (FN): Predicted ”no”; are ”yes”. (A ”Type II error.”)

The following terms are expressed in rates:

- Accuracy Rate: Overall, how often is the classifier correct? - (TP+TN)/total = (100+50)/165 = 0.91

- Misclassification Rate (Error rate): Overall, how often is it wrong? - (FP+FN)/total = (10+5)/165 = 0.09. This is equivalent to (1 - Accuracy).
- True Positive Rate (also called Sensitivity or Recall): When it's actually yes, how often did it predict yes? - TP/actual ”yes” = 100/105 = 0.95.
- False Positive Rate: When it is actually ”no”, how often does it predict ”yes”? - FP/actual ”no” = 10/60 = 0.17
- True Negative Rate: When it is actually” no”, how often does it predict ”no”? - TN/actual ”no” = 50/60 = 0.83 .This is (1 - False Positive Rate), and also known as ”Specificity”
- Precision Rate: When the classifier predicts ”yes”, how often is it correct? - TP/predicted ”yes” = 100/110 = 0.91
- Prevalence: How often does ”yes” actually occur in the sample? - Actual ”yes”/total = 105/165 = 0.64

Some other terms used in classification are also:

- Null Error Rate or Base Rate: How often would the classification be wrong if for every case it just predicted the majority class? In this example, the null error rate would be 60/165=0.36 because if always ”yes” was predicted, one would only be wrong for the 60 ”no” cases. This is a useful baseline metric. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate. This will only happen by highly unbalanced samples (actual ”yes” or ”no” is much smaller than predicted ”yes” or ”no” (Accuracy Paradox). In that case the F-score is a better gauge.
- Cohen's Kappa: In general statistics, it is a measure of the agreement between two judges (raters) about classifying items into mutually independent classes.There it is a measure of the accuracy and reliability in a statistical classification. In Machine Learning, the raters are the null error rate and the best classification, and the metric measures of how well the classifier performed as compared to a classification by chance from the Null Error Rate. A model will have a high Kappa score if there is a big difference between the Accuracy and the Null Error Rate. This use of Cohen's kappa is opposite to its use in regular statistics in how much two human raters agree in their assessment of subjective judgements.
- F Score: This is the harmonic mean (the mean of rates) of the true positive rate (recall) and the precision rate.
- ROC Curve: A commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as the threshold for assigning observations to a given class is varied.