The effect on LinkedIn profile picture on the chance to work for Apple

Thanakij Wanavit

Department of Business Management, Webster University

Course ID: BUSN 6140

ID: 4126061

Outline

1. Introduction
    1. Problem for investigation
    2. Purpose of study
    3. Significance of the study
    4. Research Question
2. Literature Review
    1. Impacts of visuals on opportunities
    2. Social media profile on mental state
    3. Exploring embedded data in images
3. Methodology design
    1. Methods
    2. Sampling
    3. Implementation
    4. Data analysis
    5. Limitations and delimitations
        i. Limitation
        ii. Delimitation
4. Significance of the study
5. Presentation and analysis of data
    1. Data
        i. Data Collection
            1. Collection process

        ii. Data Statistics
            1. Apple
            2. Non- Apple
            3. Control (US Average)
    2. Image Analysis
        i. Image general features
            1. Average brightness
            2. Filtration for non-facial images
    3. Extracted embedded Emotions
        i. Emotions of non-apple
        ii. Emotions of Apple
        iii. Comparison of emotions
    4. Analysis of the underlying emotions
        i. Reasons behind the differences
        ii.
6. Conclusion
    1. Conclusion statement
    2. Summary
    3. Implications and Significance
    4. Evaluation
    5. Further research

1. Introduction

Human beings are social animals who rely on interpersonal interaction for success as a species. For the reason stated, it is not difficult to see how each individual look can have a significant impact on their opportunities, wellbeing, and how they behave in life in general.

In the early 21st century, social media has transformed the way people communicate and how the whole social interaction is defined. (Deshpande, & O'Brien, 2019) Multiple studies has shown that social media profiles are becoming more and more representative of the mental and physical state of individuals. (Gamon & Counts, 2013; Peek et al., 2015)

1.1 Problem for investigation

The problem is how the features of a Linkedin profile picture affects the probability of the person working for Apple in the Bay area San Francisco.

1.2 Purpose of the study

The purpose of the study is to understand the reason behind the relationship of person who works for Apple and their social media profile features.

1.3 Significance of the study

The study intends to explore the psychological insights into whether the hiring practice of a company such as Apple contains biases based on individual profile picture. Although there are many studies on racial, ethnicity, country of origin, and gender on hiring practice, such as those by (Adamitis, 2000; Bendick & Nunes, 2012; Neckerman & Kirschenman, 1991) . However, there are few researches into the bias involving psychological hiring bias based on factors including emotional state, self-representation, and personal relationships. One of the reasons to believe that there is a significant is the way companies gather data about their potential hiring. There are evidences that LinkedIn and other social media profiles play significant role in companies decision.(Zide et al., 2014)had shown that employees in different industries do have significant differences in their profiles while (Chiang & Suen, 2015) has shown that people configure their social media profile differently when they are looking for a job.

1.4 Research Question

How does emotions encapsulated in a Linkedin profile picture of a person affect the person's chance to work at Apple in the Bay area San Francisco?

2. Literature review

2.1 The impacts of visuals on opportunities

2.1.1 Human history of discrimination

Discrimination is a subject of great extant. Since early in human civilization, humans have enslaved, tortured, and discriminated each other in many ways. As late as 1833, it was legal to enslave humans who are visually black. This ended when the English parliament passed Slavery Abolition Act, (1833) which took effect on 1 August 1834 ("Slavery Abolition Act | History & Impact", 2020)

2.1.2 How appearance affects opportunities and success

2.1.2 How appearance affects opportunities and success

There are multiple studies on the very broad question of how visual appearance may affect opportunities. People are treated differently depending on their appearance since they were children (Hildebrandt, 1982). Caregivers give more attention to infants that were perceived as cuter (Hildebrandt & Fitzgerald, 1981). Teachers rated attractive students to have higher IQ, better peer relations, and higher educational potential (Clifford & Walster, 1973; Clifford, 1975).

Research has found that appearance is one of the most important factors in employee selection for a wide variety of jobs. For example, Black applicants are preferred for lower status or “Black-typed” jobs (Terpstra & Larsen, 1980; Stewart & Perlow, 2001). Attractive applicants are perceived to be more qualified than their unattractive counterparts (Cash *et al*., 1977; Drogosz and Levy, 1996; Jackson *et al*., 1995; Marlowe *et al*., 1996).

Various other opportunities such as being in a sports team, music choir, and even a specific school does depend on the appearance of the subject. This, in turn lead to the income discrimination based on appearance. One study by Hamermesh & Biddle (1994,1174-1186) concluded that "plain" people earned five to ten percent less than "average-looking" people, who in turn earned five percent less than "good-looking" people. Appearance has also been found to affect the opportunities and the career path taken with a statistical significance. (Adamitis, 2000, 3-30).

Other than opportunities to learn and find occupation, when getting financing, doing business, and transacting with other people, appearance plays a vital role for obtaining acceptance. The fact that appearance is highly correlated with where a person is from, culture, language, and accent, means that they are prone to being stereotyped by others which leads to discrimination consciously or unconsciously.

2.2 Social media profile on mental states

2.2.1 Relationship between mental state to social media activities

There has been a wealth of research on using social media as a tool for determining public wellbeing including the spread of flu symptoms (Sadilek et al, 2012), building insights about disease using twitter post (Paul & Dredze, 2011). In addition to physical disease detection, there has been interesting research on mental disease including Kotikalapudi et al., (2012) where web activity of college students is analyzed for depression. Moreno et al., (2011) has demonstrated that status updates on Facebook could reveal symptoms of depression. The fact that these models exist infer that social media profile and activities have a deep relationship with one’s mental wellbeing.

2.2.2 Profile pictures on different social media outlets

In the modern world, social media has obtained a level of importance beyond just simple personal relationships. Different social media outlet has been built for a specific purpose for instance Linkedin is optimized for sharing professional profile to potential employers and colleague, github is built for sharing coding work with other developers or potential employees, twitter is for sharing opinions to the public, and stackoverflow is for sharing problems and solution to programming problems. Individuals usually post a very different kind of profile picture on each social media platform.

It has been found that most profiles are carefully manipulated for a desirable self-presentation in order to achieve a specific goal.(Larrimore, Jiang, Larrimore, Markowitz & Gorski, 2011,22)

Popular presses has linked profiles of those in dating sites to loan sites (Frier, 2009, 6b; Mogul, 2007, 8; Quinn, 2008)

2.2.**3** Drivers of social media usage

The need to belong in a society has a significant effect on relationship building (Baumeister & leary, 1995) This is a major motivator use a social media platform. A popular platform such as Facebook can be an effective way of building social connections (Sheldon, Abad & Hirsch, 2011) by enabling peer acceptance and relationship development (Yu, Tian, Vogel, & Kwok, 2010) and improving self-esteem (Gonzales & Hancock, 2011; Steinfield, Ellison, & Lampe, 2008).

2.2.4 Self presentation through Social media

Social media are usually used to accomplish self-presentational goals such as posting contents about activities that one wants to associated with (Zhao, Grasmuck, & Martin, 2008). Research has shown that popularity-seeking users tend to disclose more information on Facebook (Christofides, Muise, & Desmarais, 2009; Utz, Tanis, & Vermeulen, 2012), in order to promote their profiles (Utz et al., 2012). Social media profile is one of the best tool to represent self-presentation goals of individuals (Back et al., 2010).

2.3 Exploring embedded data in images

2.3.1 Object detection

Humans glance at an image and instantly know what objects are in the image (Redmon, et al., 2016). Object detection (Girshick et al., 2014; Viola & Jones, 2001)(Viola, P., & Jones, M. (2001) is one of the most common way to extract data from an image, such systems are able to identify a specific instance of a class, in contrast to classification algorithm where the goal is to understand the difference between objects of the same class (Murase Nayar, 1995). In order to understand a social profile picture, it would make sense to use an object detection in conjunction with classification algorithm such as ones by Papageorgiou, C. (2000).

Implementing an object detection system

The most common approach (Rainer Lienhart, 2002) to tackling this object detection is to re-purpose existing trained classifiers to assign labels to bounding boxes in a scene. For example, a standard sliding window approach (Girshick et al., 2014) can be used where a classifier determines the existence of an object and its associated label for all possible windows in the scene. Though effective, this type of algorithm requires a lot of computational resources which render them impractical for many circumstances. Modern algorithms such as deep neural networks (DNN) have shown superior performance in a range of different applications (Krizhevsky, A. and Sutskever, I., Hinton, 2012; Simonyan & Zisserman, 2015), with object detection being one of the key areas where DNNs have significantly outperformed existing approaches such as Sparse Coding (Deng et al., 2012) and sift(Sánchez & Perronnin, 2011).

Yolo real time object detection

You Only Look Once (YOLO) object detection approach (Redmon et al., 2016) was proposed that mitigated the computational complexity issues associated with Region-CNN(R-CNN)(Girshick et al., 2014) by formulating the object detection problem as a single regression problem, where bounding box coordinates and class probabilities are computed at the same time. YOLO has demonstrated a significantly higher speed and lower computational resource requirements over R-CNN while providing a satisfactory result.

Pascal VOC 2012 Dataset

The Pascal VOC 2012 dataset is the most widely used dataset and challenges to train and test object detection networks and measure the performance relative to others. It includes a 1.9 GB of training/validation and 1.8 GB of testing dataset for 1000 different categories of objects. It is common to use models which are pre-trained on this dataset as a base for transfer-learning in order to classify images on other categories including emotions. (M. , Eslami, S. M. A. , Van Gool, L. , Williams, C. K. I. , Winn, J. and Zisserman, A. 2012)

2.3.2 Features Classification

2.3.2.1 Face features classification

Pose-invariant face recognition can be divided into 4 groups: pose-robust feature extraction, multi-view subspace learning, face synthesis based on 2D methods, and face synthesis based on 3D methods. One of the ways to extract features is by using the symmetric interpolation for self-occluded regions (Méndez N, Bouza LA, 2017)

While all the methods are valid and are still used in many cases, neural network allows all of the methods to be performed simultaneously using GPU acceleration (general co-processor unit). This result in the benefits of all the methods without a significant overhead.

2.3.2.2 Classification with neural network

For each face recognition method in 2.3.2.1, each feature is generated programmatically and given a score. Collectively, these features should summaries all the useful information in the image while reducing the amount of data fed into the final layer of image classifier, usually a fully connected function with SoftMax activation function (figure 2.3.2.1) which outputs a probability of a classification result. In Simonyan, K., & Zisserman, A. (2015) 's case, it represents the probability of the object containing each of the 1000 pre-trained categories including dog breeds. (K., & Zisserman, A. 2015, p 13). Although these 1000 pre-trained category is only useful for classifying the 1000 classes, a transfer learning method can be used to classify different images into different features as demonstrated by Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). Pre-trained models are very useful because it is possible to transfer these networks to use in another problem without starting from scratch, speeding up the development of other solutions significantly.

$$\mathbf{softmax}(z)_i = \frac{\exp(z_i)}{\sum_{l=1}^{k} \exp(z_l)}$$

Figure 2.3.2.1 Softmax activation function (Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. 2011)

2.3.2.3 Visualising Neural network layers in an image

The method outlined by Graetz, F. M. (2019, July 21) can be used to visualize and plot the locations of the image that has an effect on output feature node of interest(figure 2.3.2.2). This allows us to give it a human description of the node, for example mouth, forehead, hair color.

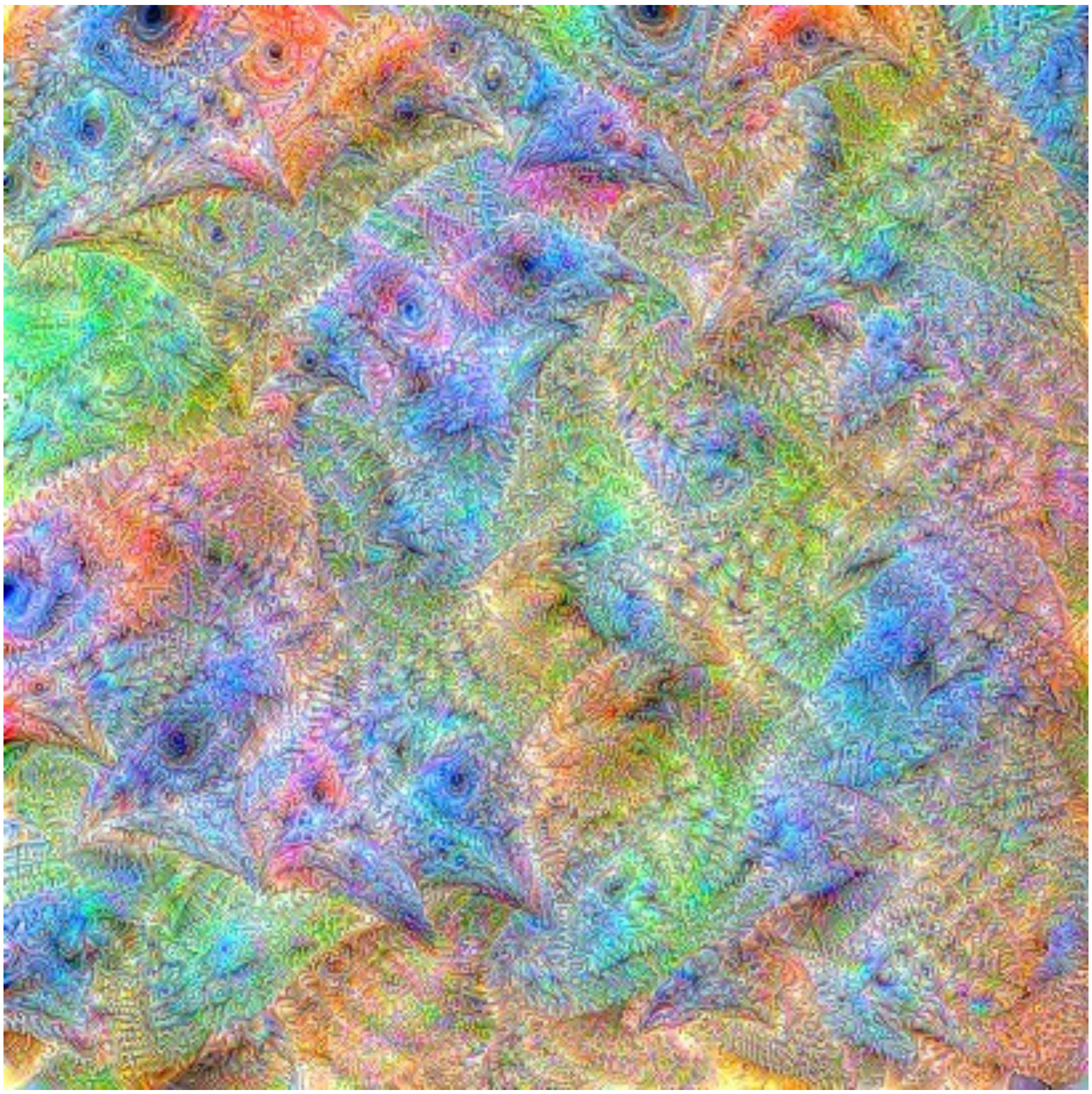


figure 2.3.2.2 Plot of an eye neural network node (Graetz, F. M. 2019, July 21)

2.3.3 Combining Object detection and Classification Methods

Extraction of a deeper information inn an image is possible using a combination of a trained object detection DNN followed by classification DNN. This is demonstrated by an emotion extraction algorithm such as Emopy (Perez, A. 2018).

Different configurations and variation of Convolutional Neural Network has been tested by researchers to extract the data from the images. These results, although usually highly specific for one set of prediction, e.g. determining if an image contains a specific species of dog, can be transferred to other models with completely different goals using the transfer learning technique (Yang, Zhang, Dai, & Pan, 2020).

One of the pre-trained model available in EmoPy is the Facial Expression Recognition (FER) model which is a convolutional network trained on the FER dataset by Barsoum et al., (2016).

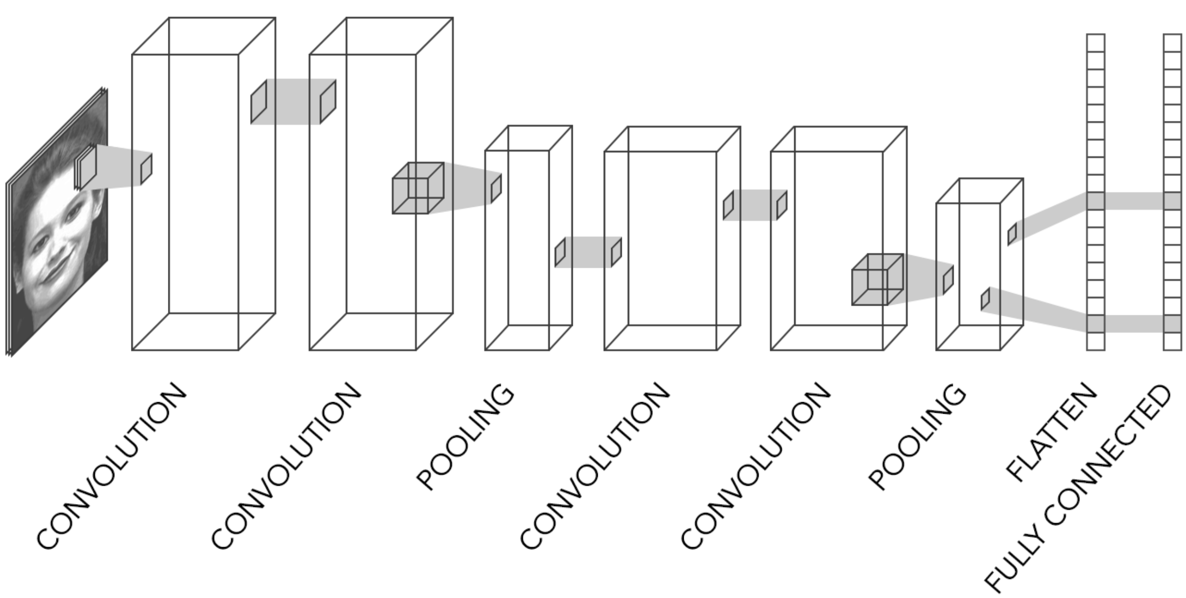


Figure 2.3.3.1 An example of convolutional Neural network (Perez, A. 2018)

2.3.3.1 FER Dataset

The FER+ annotations provide a set of new labels for the standard Emotion FER dataset. In FER+, each image has been labeled by 10 crowd-sourced taggers, which provide better quality ground truth for still image emotion than the original FER labels. Having 10 taggers for each image enables researchers to estimate an emotion probability distribution per face. This allows constructing algorithms that produce statistical distributions or multi-label outputs instead of the conventional single-label output (Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. 2016)

3. Methodology Design

The study aims to collect the data between occupation and face features of each male from linkedin in the region of SanFrancisco with the company they work for.

3.1 Methods

3.1.1 Obtain the data from website

- Use browser "Google Chrome" with a webdriver of "chromedriver"
- From the website www.linkedin.com, navigate to search for people page as shown in Fig 3.1.1
- Select current companies from the list of 20 most common companies (fig 3.1.1.2)
- Save this picture and company name in a database

- Collect 100 samples from each company

A video demo of the following task can be viewed at

https://www.youtube.com/watch?v=N3sfYo867Sk

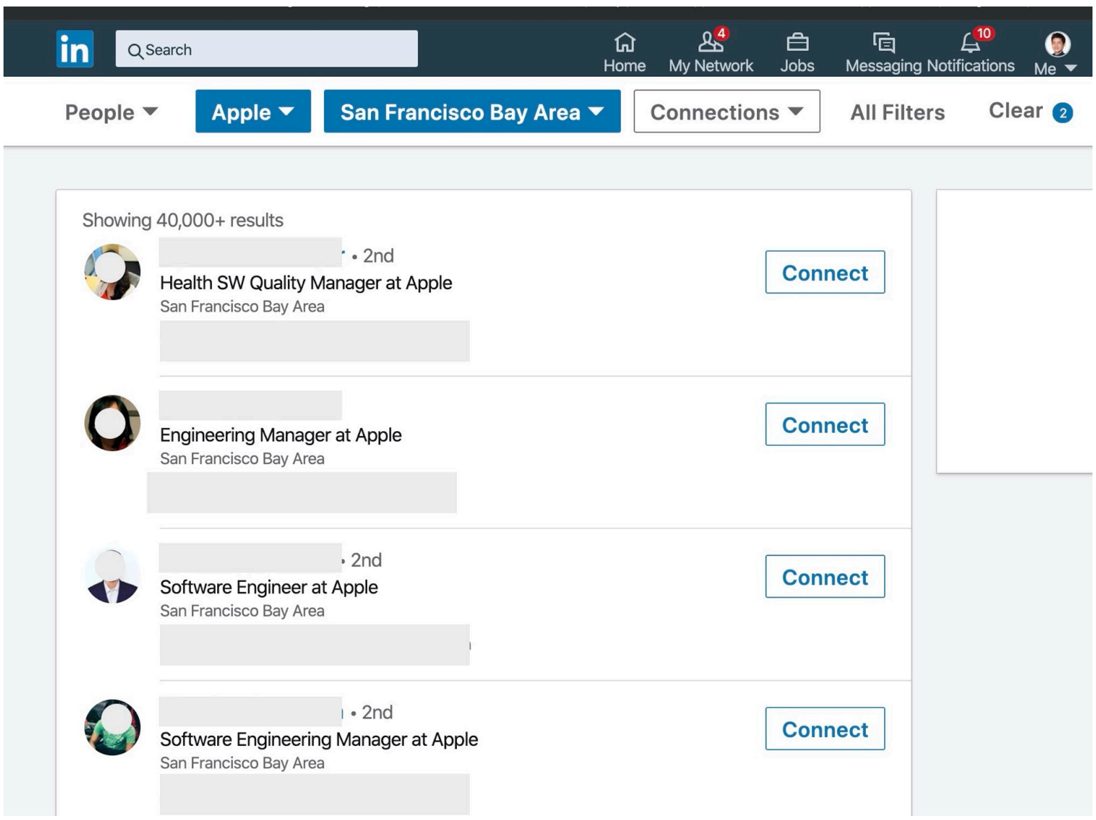


Fig 3.1.1 Search for people page (www.linkedin.com, 10 Feb 2020)

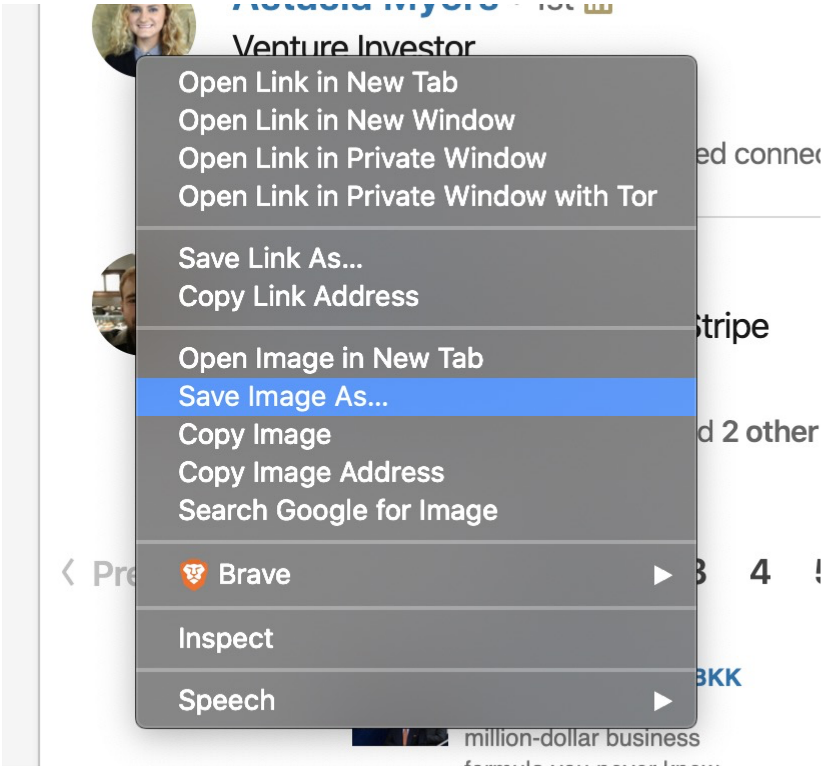


Fig 3.1.1.2 save image from profile picture within the linkedin search page (www.linkedin.com, 10 Feb 2020)

3.1.2 Extract emotional feature from each face

- Using a package management tool "pip" to install all the packages below
- Install EmoPy library(Perez, A.,2017) and its prerequisite from the official repository
- Import pre-trained FER model (Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. ,2016)
- Use the script in Appendix 7.2 to obtain the probability of the emotion encapsulated within the image. The possible emotions are listed in figure 3.1.2.1

```
['calm', 'anger', 'happiness', 'surprise', 'disgust', 'fear', 'sadness']
```

Figure 3.1.2.1 list of emotions of interest

- Scores are recorded for each picture
- Source code for the task can be found in appendix (7.3.1.2)

A video demo of the following task can be viewed at

https://www.youtube.com/watch?v=PFrGgnHgdRQ

3.1.3 Features analysis

- The difference between the average score for each feature for apple when compared to other company represents the degree of bias.
- The standard deviation of each feature within Apple should represent the similarity of the hiring choice.
- For each emotion generated, plot and output of the generated feature and compare between he 2 groups.
- This should indicate which part of the profile image feature is important in determining career

3.2 Sampling

- Systematic sampling using 5 separate robots with an account in Oregon to collect 100 pictures systematically every 5 profiles it sees.
- The algorithm will filter for only employees with computer-science related jobs using the LinkedIn filter.
- Sample size is 200 Pictures

3.3 Instrumentation

- Data will be collected using selenium(SeleniumHQ, 2019) controller which controls google chrome browser on a cloud server using aws ec2 instance t3.micro.
- Data collected will be uploaded to s3 bucket for storage and the uri will be collected in a dynamodb database.(Rangel, 2015)

3.4 Data Analysis

- Score on each emotion listed in Figure 3.1.2.1 are obtained and put into a single database
- The score for each emotional feature in each group (Apple or others) is analyzed for standard deviation, covariance, average.
- The most important face feature is the one which has the highest standard deviation when comparing employees from different companies, but low standard deviation when comparing the employees from the same company
- The final outcome will be represented as heatmap of correlation between features and companies as in fig 3.4.1

3.5 Control group

- A control group is used in order to make sure that the Emopy algorithm does give a different result for different kind of profile picture.
- 350 of the pictures of with a filter location of USA is collected using the same methods as Apple employees.
- The control group is analyzed for differences between apple and non-apple software engineers in Bay Area SanFrancisco

3.6 Random forest model for multidimensional analysis

Random forest model is used to analyse the deeper relationship of the data in order to take into account all dimensions at the same time. Due to the binary nature of the data, (https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d)

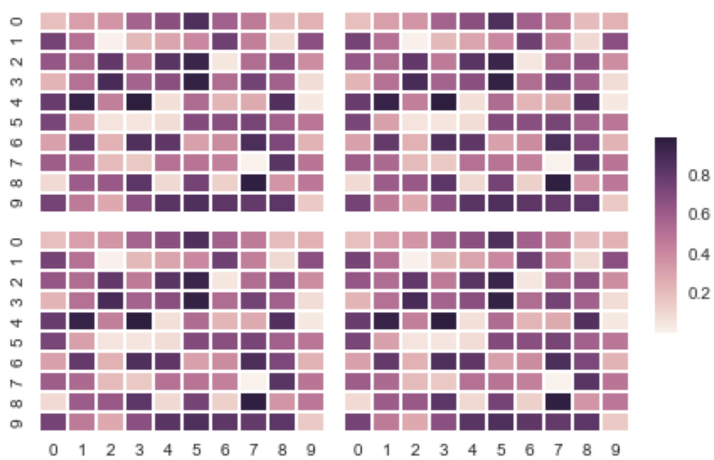


Fig 3.4.1 Generic heatmap diagram generated with seaborn 0.10.0 (https://seaborn.pydata.org/, 10 Feb 2020)

3.5 Limitations and Delimitations

3.5.1 Limitations

A Company like Apple hires personnel in many different functions from cleaning, reception to business managers. These groups of individuals are likely to display varyingly different behaviors.

LinkedIn ranks profiles based on various information including the source of IP address, relationship of the requester to the profiles. This could cause a bias in the samples selected.

3.5.2 Delimitations

Only Subjects that are in the occupation directly related to computer science are selected because this group is the most common employee in the Apple headquarter. Other occupations may display a different behavior.

Pictures should be taken from the LinkedIn profile of people who are located in the Bay area San Francisco, California and have opened their profile to the public. This group of subjects in a narrow area can be assumed to share many characteristics and behaviors.

To reduce bias, a robot logs in from multiple locations and create 5 LinkedIn profiles to do so. The IP in Oregon which is the closest zone accessible by AWS server are used in order to reduce the effect of LinkedIn algorithm prioritizing subjects with relationship to source IP location.

5. Significance of the study

The study will allow companies to develop and improve their tools for better decisions based on how they select candidates. Biases racially, emotionally, and morally.

For candidates who would like to apply for jobs, this method may give an insight to the chance of being accepted at a specific company.

6.Analysis and findings

6.1 Results

6.1.1 Overall Results

The data collected in arranged into a SQL table as shown in 6.1.1.1. The data is then analyzed by splitting into 3 groups; Software Engineer at Apple, Software Engineer Non-Apple, and Control (US-Average). Each group is given 2 boolean labels as shown. The summary histogram in 6.1.1.2 shows that the profile picture images are showing the same pattern trend respectless if they work for apple. However, this does not mean that there is no differences. More details are explored when comparing a higher dimension spaces in the next section.

6.1.1.1 General Statistics

The number of samples analyzed as shown in fig 6.1.1.4 shows that the we have collected 851 samples of apple, 416 samples of other companies in the Bay Area and 517from the general US population. These numbers are limited due to the work involved in data collection. More data can be collected. Fig 6.1.1.5 and fig 6.1.1.6 shows standard deviation and mean of the scores received for each sample. Although none of the columns are showing a sufficiently large variation on their own, deeper dimensional data needs to be studied to explore a deeper relationship.

isAppleEmployee	isControl	count
False	False	416.0
	True	517.0
True	False	851.0

Fig 6.1.1.4 Summary of number of samples collected for each group

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	False	6.707473	1.673548	7.554485	10.381702	1.004194	6.943185	2.201570
	True	6.830195	1.694914	7.808399	10.443877	1.023408	7.260752	2.310475
True	False	6.863180	1.732223	7.802718	10.334477	1.052713	7.499246	2.361982

Fig 6.1.1.5 Summary of standard deviation of samples collected for each group

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	False	17.083514	20.721148	21.234029	10.371244	1.275987	20.287157	9.026921
	True	16.667585	20.610775	20.557491	10.975039	1.241604	20.755076	9.192431
True	False	17.012138	20.728772	19.825168	10.246094	1.293032	21.579587	9.315207

Fig 6.1.1.6 Summary of standard deviation of samples collected for each group

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	False	2.546937	12.381570	2.810784	0.998993	1.270658	2.921880	4.100219
	True	2.440279	12.160366	2.632741	1.050859	1.213205	2.858530	3.978590
True	False	2.478754	11.966570	2.540803	0.991448	1.228285	2.877568	3.943810

Fig 6.1.1.6 Summary of mean/standard deviation of samples collected for each group

	surprise	calm	happiness	sadness	disgust	anger	fear	isAppleEmployee	isControl
0	6.424556	18.838278	22.863583	29.152518	3.725452	9.151204	9.844409	True	False
1	23.759138	20.908619	8.609241	0.000000	2.561888	30.243663	13.917451	False	True
2	20.838966	21.480569	17.229608	0.000000	2.151285	26.642347	11.657226	True	False
3	23.837943	24.728503	26.193952	0.000000	0.391783	16.147272	8.700548	False	False
4	12.898688	17.326420	14.220505	16.861725	0.074804	28.076196	10.541662	True	False

Fig 6.1.1.1 Mean score of the first 5 candidates in the database

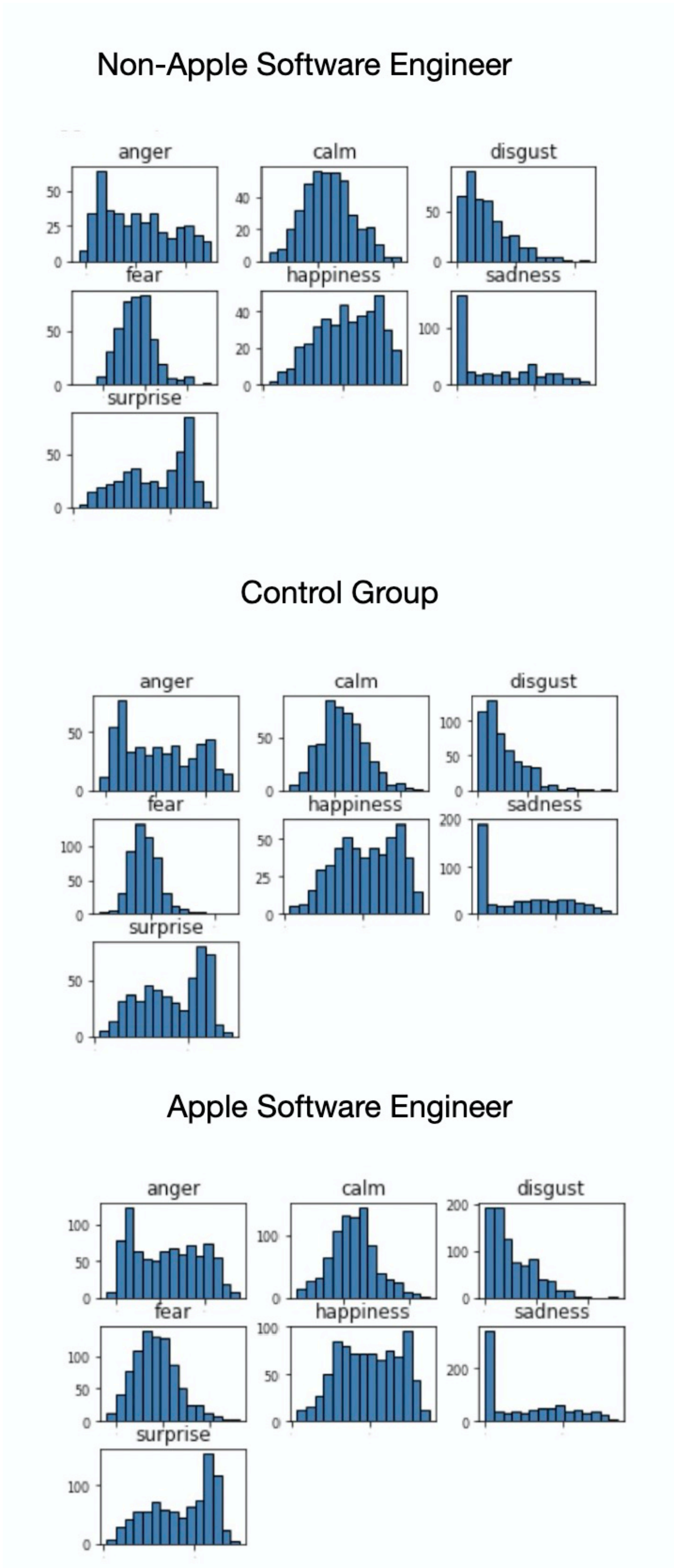


Fig 6.1.1.2 Summary histogram over all Emotions for each group of samples

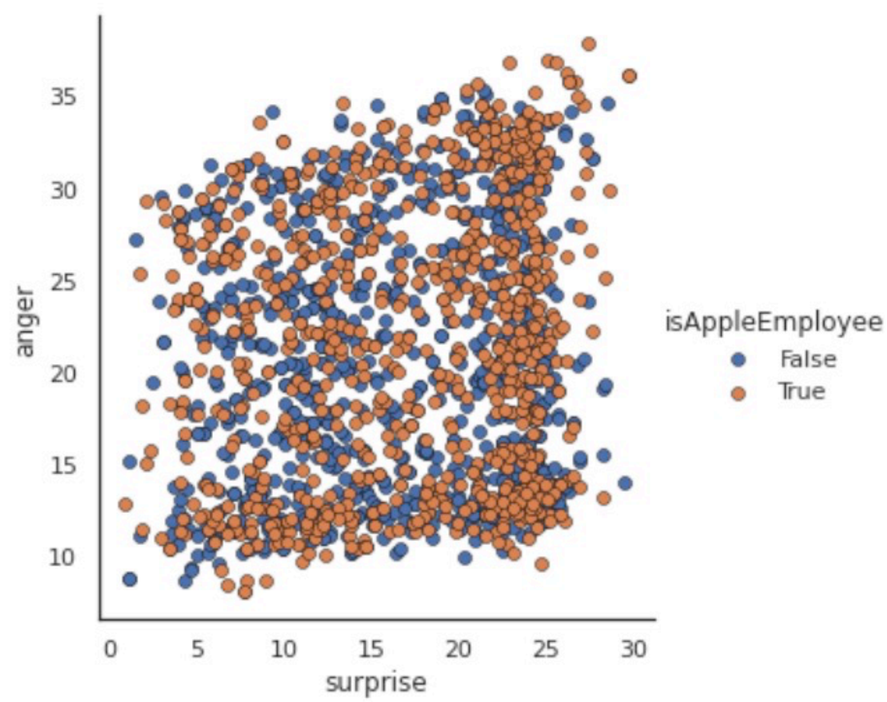


Fig 6.1.1.3 Scatter plot between anger and surprise comparing apple employee and general population

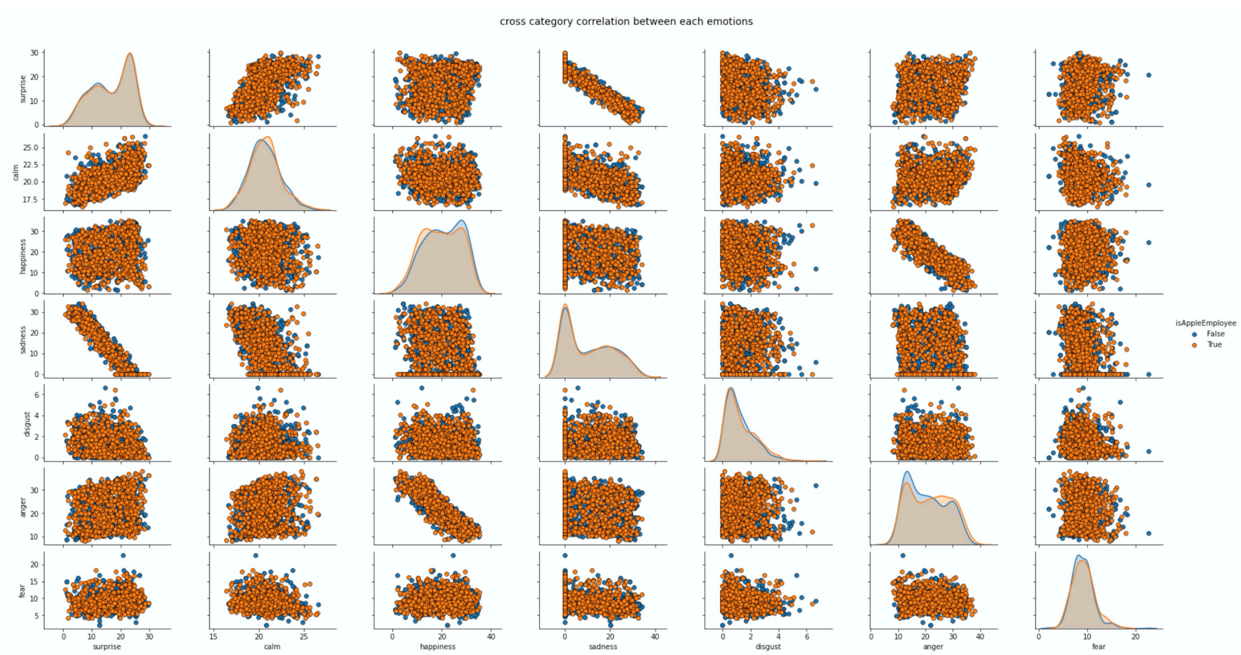
6.2 Comparisons between software engineers who work for apple and other company in the bay area

6.2.1 general Result for the group

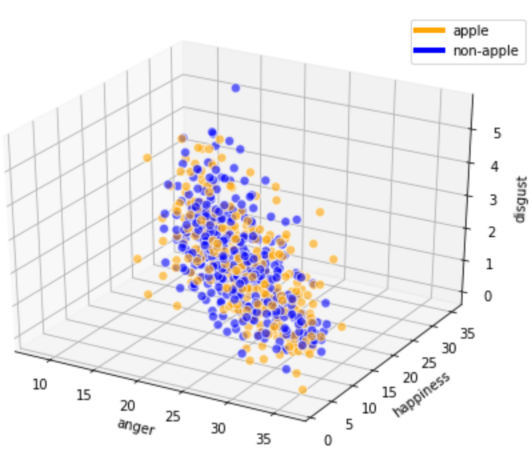
Figure 6.2.1 shows a table of comparison between each category in software engineer in Apple and out of apple. It is clear that there are some differences in There is a small difference in the mean between happiness disgust and anger however, they are far from significant.

6.2.2 Comparison in 3 dimensions

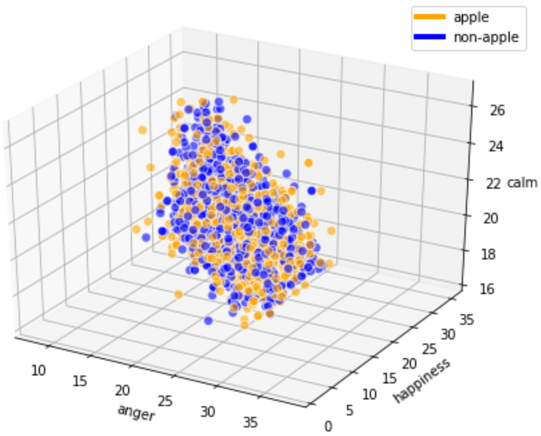
Figure 6.2.2 and 6.2.3 shows that although the features that has the largest differences in distribution ie anger, happiness, and calm are plotted, there is no clear pattern to distinguish the apple and non-apple employees. This suggests that these groups may share much of the same emotion when presenting their profile picture in general



6.2.1 Cross correlation between each category of emotion for apple versus others for software engineers



6.2.2 3D plot for emotions [anger, happiness, and disgust] for the Bay area samples



6.2.3 3D plot for emotions [anger, happiness, and calm] for the Bay area samples

6.2.3 Comparison in a deeper dimension

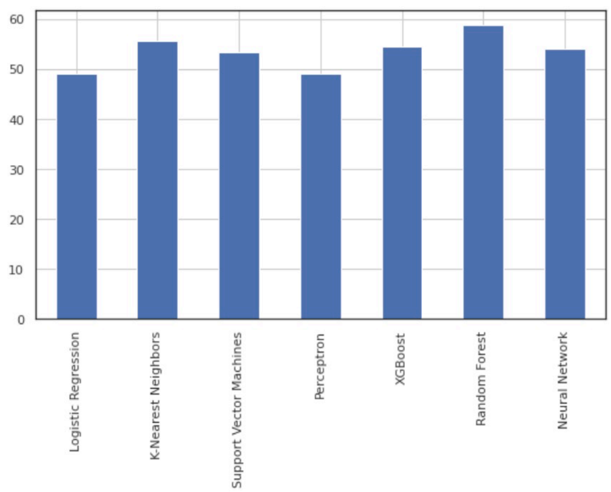
Machine learning algorithm can be used to determine a deeper relationship. Data is split into 2 sets, training and testing, with 80:20 ratio. After training, each of the machine learning model is tested against the test data. The number of data used is 416 due to some machine learning models being optimized for a balanced group samples

6.2.3.1 Machine Learning models result comparison

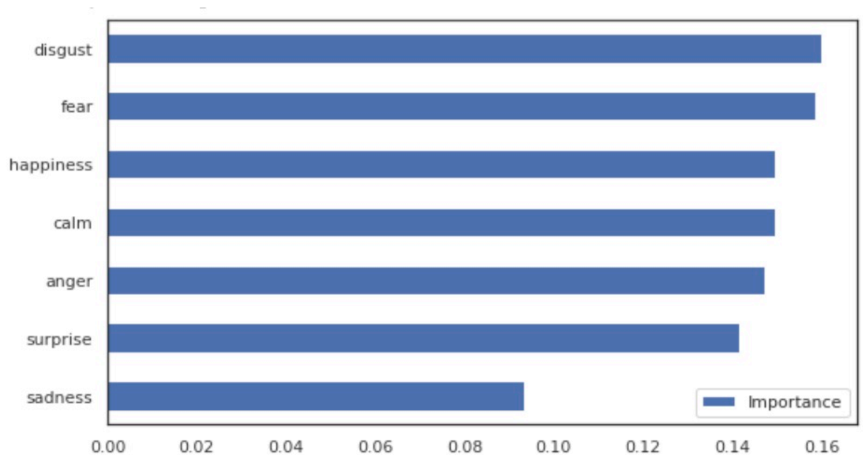
Fig 6.2.1.3.1 shows that most models perform better than 50%. Random forest model is able to forecast with 59% accuracy which is significantly more than 50%. We are able to conclude that there is a deep relationship between the two samples. If a neural network is optimized for the task, and more samples are collected, the model should be able to predict the group with a much higher accuracy.

6.2.3.2 Importance of each emotion

Random forest model is the best model in the experiment therefore, each feature is plotted for their contribution in the forecasting model. Fig 6.2.1.3.2 shows that disgust and fear emotions are the most important when trying to determine the group of samples.



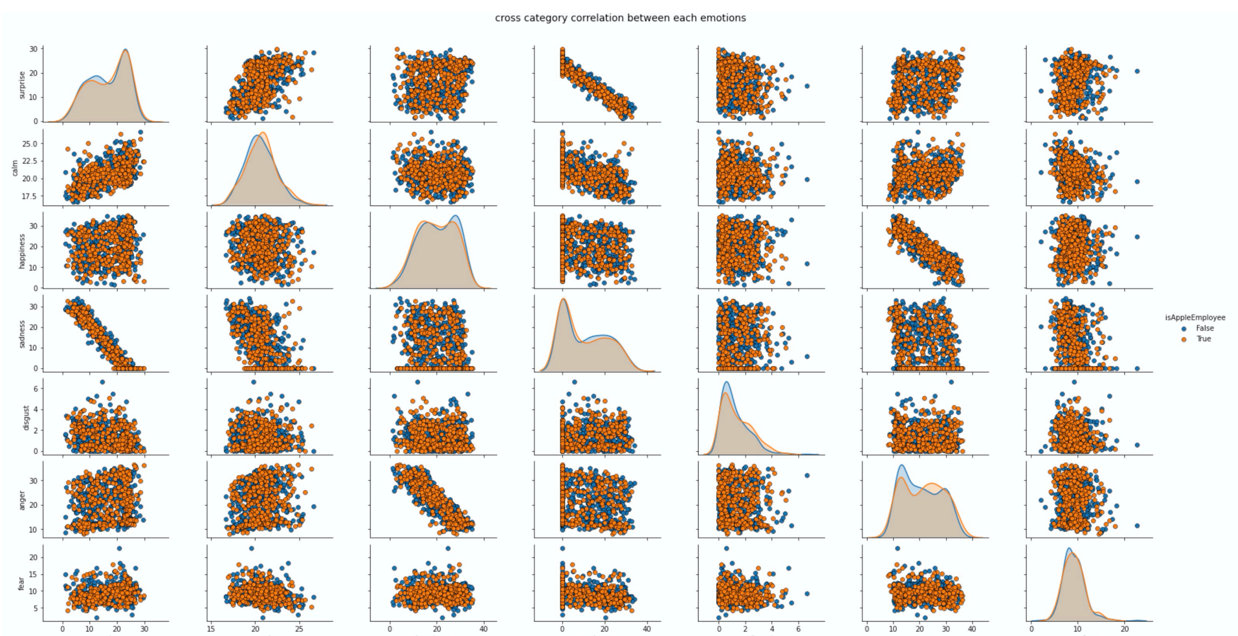
6.2.3.1 Prediction score for each machine learning model



6.2.3.2 Shows the importance of each emotion based on Random forest model used in Fig 6.2.3.1

6.3 Comparison between Apple software engineer and US average person

Although there is a small difference between each group of subjects when each individual category shown in the pair-plot in fig 6.2.1, the multidimensional relationship is reasonable strong. A random forest model is able to predict with 62% accuracy. This signifies that there is a significant deep relationship between apple and control group employees when data is looked at in 7 dimensions. Fig 6.2.5 shows that Anger and disgust are the best predictor at 16% and 15% contribution which matches with the distribution difference shown in the pair-plot. However, every single emotion contributes to more than 10 % of the prediction.



6.3.1 Cross correlation plot between apple employees and American average person

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	**False**	6.707473	1.673548	7.554485	10.381702	1.004194	6.943185	2.201570
	True	16.424275	3.442533	21.124895	30.276228	2.799079	18.943167	6.257418
True	**False**	6.924011	1.747234	7.735185	10.365619	1.105743	7.560401	2.303708

6.3.2 Standard deviation of the emotion score for each category, control is the us average population

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	**False**	17.083514	20.721148	21.234029	10.371244	1.275987	20.287157	9.026921
	True	43.919258	55.329613	55.328776	30.998598	3.368369	55.623761	24.711452
True	**False**	16.983058	20.756385	19.814279	10.275493	1.382727	21.548557	9.239502

6.3.3 Mean emotion score for each category, control is the us average population

```
# Random Forest
random_forest = RandomForestClassifier(n_estimators=500)   # instantiate
random_forest.fit(X_train, Y_train)                         # fit
acc_rf = random_forest.score(X_val, Y_val)                  # predict + evaluate

print('Random Forest labeling accuracy:', str(round(acc_rf*100,2)),'%')
```

```
Random Forest labeling accuracy: 62.17 %
```

6.2.4 Random forest is forecasting with 62.17% accuracy

```
# XGBoost, same API as scikit-learn
gradboost = xgb.XGBClassifier(n_estimators=1000)            # instantiate
gradboost.fit(X_train, Y_train)                              # fit
acc_xgboost = gradboost.score(X_val, Y_val)                  # predict + evalute

print('XGBoost labeling accuracy:', str(round(acc_xgboost*100,2)),'%')
```

6.3.5 XgBoost model forecasting with 62.17%. accuracy

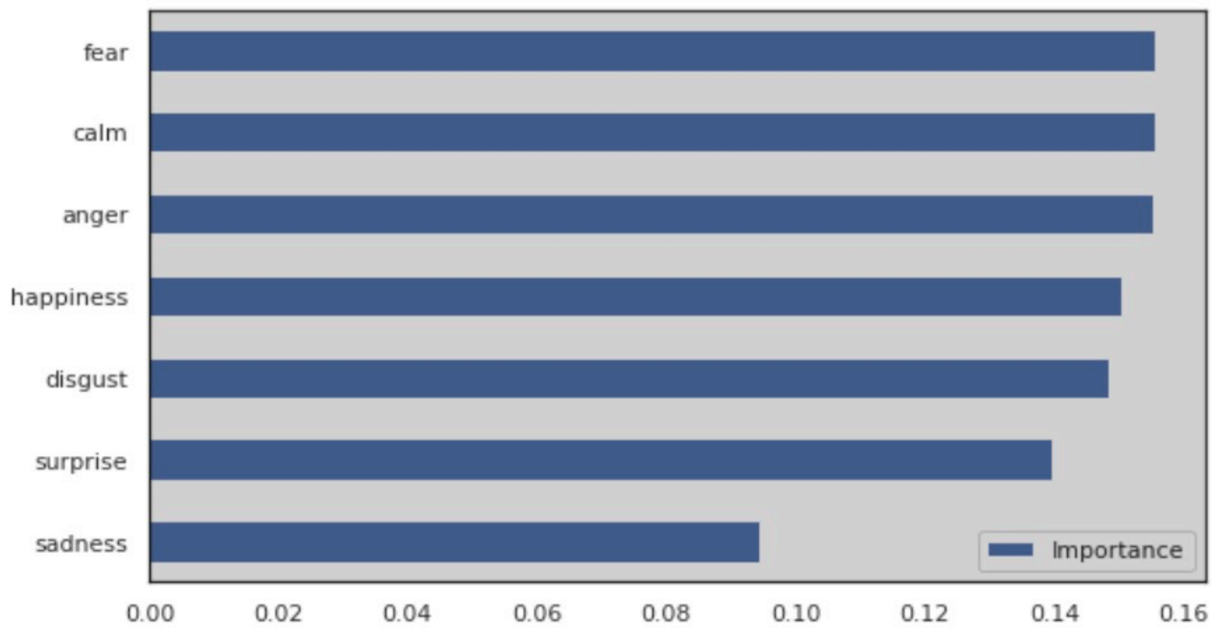


Fig 6.3.6 The contribution of each factor to the Random Forest model

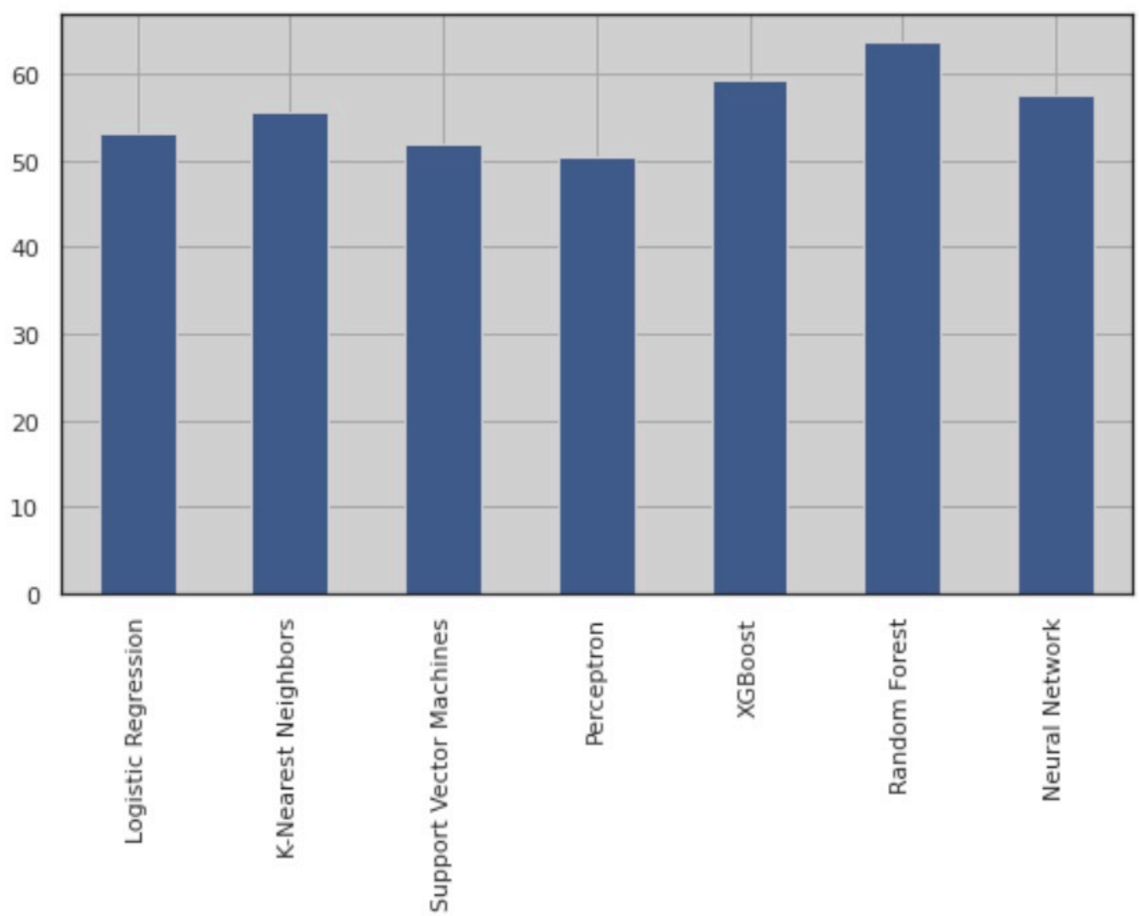


Fig 6.3.7 Forecast comparison by various machine learning models. More than 50% indicates relationship between Y and X data.

7. Conclusion

There is a deep but unclear relationship between facial expression on the profile page and the probability of a person working for Apple. Although it is difficult to pinpoint what is the difference, clustering algorithms suggests that there is a significant correlation. These results suggest that the difference may be due to other aspects of facial features which has nothing to do with the emotions. More studies with a deeper and different kind of network on the dataset can help to clarify the relationship.

8. References

Adamitis, E. (2000). Appearance Matters: A Proposal to Prohibit Appearance Discrimination in Employment. Washington Law Review, 75(1), 3-30.

Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. (2016). Training deep networks for facial expression recognition with crowd-sourced label distribution. ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction, 279–283. https://doi.org/10.1145/2993148.2993165

Brittanica (2020) Slavery Abolition Act | History & Impact.. Retrieved 26 January 2020, from https://www.britannica.com/topic/Slavery-Abolition-Act

Deshpande, A. K., Deshpande, S. B., & O'Brien, C. A. (2019). Hyperacusis and social media trends. Hearing, Balance and Communication, 17(1), 1–11. doi: 10.1080/21695717.2018.1539321

Frier, S.(2009, June 30). Peer-to-peer lending is flourishing amid the credit crisis. Sacremento Bee,p. 6b.

Gatewood, R., Lahiff, J., Deter, R., & Hargrove, L. (1989). Effects Of Training On Behaviors of The Selection Interview. Journal Of Business Communication, 26(1), 17-31. doi: 10.1177/002194368902600103

Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep learning. Cambridge, MA: The MIT Press.

Guido, G., Pichierri, M., Pino, G., & Nataraajan, R. (2018). Effects of Face Images and Face Pareidolia on Consumers Responses to Print Advertising. Journal of Advertising Research, 59(2), 219–231. doi: 10.2501/jar-2018-030

Hamermesh, D., & Biddle, J. (1994). Beauty and the Labor Market. American Economic Review, dec, 1174-1186.

Kaggle (2020).,Your Machine Learning and Data Science Community, Retrieved March 8, 2020, from https://www.kaggle.com/

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., & Krasin, I. (2017). The Open Images Dataset V4 Unified image classification , object detection , and visual relationship detection at scale.

Perez, A. (2018). EmoPy: a machine learning toolkit for emotional expression. ThoughtWorks. https://www.thoughtworks.com/insights/blog/emopy-machine-learning-toolkit-emotional-expression

Larrimore, L., Jiang, L., Larrimore, J., Markowitz, D., & Gorski, S. (2011). Peer to Peer Lending: The Relationship Between Language Features, Trustworthiness, and Persuasion Success. Journal Of Applied Communication Research, 39(1), 19-37. doi: 10.1080/00909882.2010.536844

Méndez N, Bouza LA, Chang L, Méndez-Vázquez H, editors. Efficient and Effective Face Frontalization for Face Recognition in the Wild. Iberoamerican Congress on Pattern Recognition; 2017: Springer.

Mogul, M. (2007, July). Look for a higher return? Consider lending to a peer. Kiplinger Business Forecasts, 8. http://www.kiplinger.com/businessresource/forecast/archive/consider_lending_to_a_peer_070709.html.

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830

Perez, A. (2017). EmoPy (0.0.5). ThoughtWorks Arts. https://github.com/thoughtworksarts/EmoPy

Perez, A. (2018). EmoPy: a machine learning toolkit for emotional expression. ThoughtWorks. https://www.thoughtworks.com/insights/blog/emopy-machine-learning-toolkit-emotional-expression

Quinn, J. (2008, August 27). Financing is as close as the nearest angel. Retrieved from http://www.bloomberg.com/apps/news?pid=20601039&sid=a7LI2Jh3XdHM

Rangel, D. (2015). DynamoDB: everything you need to know about Amazon Web Service's NoSQL database. Retrieved from https://aws.amazon.com/dynamodb/

Redmon, Joseph, Divvala, Santosh, Girshick, Ross, … Ali. (2016, May 9). You Only Look Once: Unified, Real-Time Object Detection. Retrieved from https://arxiv.org/abs/1506.02640

SeleniumHQ. (2019, December 10). SeleniumHQ/selenium. Retrieved from https://github.com/SeleniumHQ/selenium

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–14.

Yang, Q., Zhang, Y., Dai, W., & Pan, S. J. (2020). Transfer learning. Cambridge: Cambridge University Press.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, 4(January), 3320–3328.

Graetz, F. M. (2019, July 21). How to visualize convolutional features in 40 lines of code.

Retrieved from https://towardsdatascience.com/how-to-visualize-convolutional-features-in-40-lines-of-code-70b7d87b0030

Gamon, M., & Counts, S. (2013). Letter for comprehensive pediatric nursing. Comprehensive Child and Adolescent Nursing, 36(1–2), 168–169. https://doi.org/10.3109/01460862.2013.798190

9.Appendix

7.1 Resnet - 50

NetChain[

		image
	Input	array (size: 3×224×224)
conv1	ConvolutionLayer	array (size: 64×112×112)
bn_conv1	BatchNormalizationLayer	array (size: 64×112×112)
conv1_relu	Ramp	array (size: 64×112×112)
pool1_pad	PaddingLayer	array (size: 64×113×113)
pool1	PoolingLayer	array (size: 64×56×56)
2a	NetGraph (12 nodes)	array (size: 256×56×56)
2b	NetGraph (10 nodes)	array (size: 256×56×56)
2c	NetGraph (10 nodes)	array (size: 256×56×56)
3a	NetGraph (12 nodes)	array (size: 512×28×28)
3b	NetGraph (10 nodes)	array (size: 512×28×28)
3c	NetGraph (10 nodes)	array (size: 512×28×28)
3d	NetGraph (10 nodes)	array (size: 512×28×28)
4a	NetGraph (12 nodes)	array (size: 1024×14×14)
4b	NetGraph (10 nodes)	array (size: 1024×14×14)
4c	NetGraph (10 nodes)	array (size: 1024×14×14)
4d	NetGraph (10 nodes)	array (size: 1024×14×14)
4e	NetGraph (10 nodes)	array (size: 1024×14×14)
4f	NetGraph (10 nodes)	array (size: 1024×14×14)
5a	NetGraph (12 nodes)	array (size: 2048×7×7)
5b	NetGraph (10 nodes)	array (size: 2048×7×7)
5c	NetGraph (10 nodes)	array (size: 2048×7×7)
pool5	PoolingLayer	array (size: 2048×1×1)
flatten_0	FlattenLayer	vector (size: 2048)
fc1000	LinearLayer	vector (size: 1000)
prob	SoftmaxLayer	vector (size: 1000)
	Output	class

]

Fig 7.1 Representation of Resnet50 network, for classification of image features.

https://github.com/WeidiXie/Keras-VGGFace2-ResNet50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

7.3.1.2 Source code for giving scores to emotions in images

```
from keras.models import load_model

import cv2

from scipy import misc

import numpy as np

import json

from pkg_resources import resource_filename

import imageio



class FERModel:

    """

    Pretrained deep learning model for facial expression recognition.
```

```
    :param target_emotions: set of target emotions to classify
    :param verbose: if true, will print out extra process information
    **Example**::
        from fermodel import FERModel
        target_emotions = ['happiness', 'disgust', 'surprise']
        model = FERModel(target_emotions, verbose=True)
    """
# we picked the options with the highest number of emotions based on the publicly available
dataset (FER Dataset)
    POSSIBLE_EMOTIONS = ['anger', 'fear', 'calm', 'sadness', 'happiness', 'surprise', 'disgust']

    def __init__(self, target_emotions, verbose=False):
        self.target_emotions = target_emotions
        self.emotion_index_map = {
            'anger': 0,
            'disgust': 1,
            'fear': 2,
            'happiness': 3,
            'sadness': 4,
            'surprise': 5,
            'calm': 6
        }
        self._check_emotion_set_is_supported()
```

```
        self.verbose = verbose
        self.target_dimensions = (48, 48)
        self.channels = 1
        self._initialize_model()

    def _initialize_model(self):
        print('Initializing FER model parameters for target emotions: %s' % self.target_emotions)
        self.model, self.emotion_map = self._choose_model_from_target_emotions()

    def predict(self, image_file):
        """
        Predicts discrete emotion for given image.
        :param images: image file (jpg or png format)
        """
        image = imageio.imread(image_file)
        return self.predict_from_ndarray(image)

    def predict_from_ndarray(self, image_array):
        """
        Predicts discrete emotion for given image.
        :param image_array: a n dimensional array representing an image
        """
        gray_image = image_array
```

```
        if len(image_array.shape) > 2:
            gray_image = cv2.cvtColor(image_array, code=cv2.COLOR_BGR2GRAY)
        resized_image = cv2.resize(gray_image, self.target_dimensions,
interpolation=cv2.INTER_LINEAR)
        final_image =
np.array([np.array([resized_image]).reshape(list(self.target_dimensions)+[self.channels])])
        prediction = self.model.predict(final_image)
        # Return the dominant expression
        dominant_expression = self._print_prediction(prediction[0])
        return dominant_expression


    def _check_emotion_set_is_supported(self):
        """
        Validates set of user-supplied target emotions.
        """
        supported_emotion_subsets = [
            set(['calm', 'anger', 'happiness', 'surprise', 'disgust', 'fear', 'sadness']),
            set(['anger', 'fear', 'surprise', 'calm']),
            set(['happiness', 'disgust', 'surprise']),
            set(['anger', 'fear', 'surprise']),
            set(['anger', 'fear', 'calm']),
            set(['anger', 'happiness', 'calm']),
            set(['anger', 'fear', 'disgust']),
```

```
            set(['calm', 'disgust', 'surprise']),
            set(['sadness', 'disgust', 'surprise']),
            set(['anger', 'happiness'])
        ]
        if not set(self.target_emotions) in supported_emotion_subsets:
            error_string = 'Target emotions must be a supported subset. '
            error_string += 'Choose from one of the following emotion subset: \n'
            possible_subset_string = ''
            for emotion_set in supported_emotion_subsets:
                possible_subset_string += ', '.join(emotion_set)
                possible_subset_string += '\n'
            error_string += possible_subset_string
            raise ValueError(error_string)

    def _choose_model_from_target_emotions(self):
        """
        Initializes pre-trained deep learning model for the set of target emotions supplied by user.
        """
        model_indices = [self.emotion_index_map[emotion] for emotion in self.target_emotions]
        sorted_indices = [str(idx) for idx in sorted(model_indices)]
        model_suffix = ''.join(sorted_indices)
        #Modify the path to choose the model file and the emotion map that you want to use
        if(model_suffix == '0123456'):
```

```
            model_file = 'models/conv_model_%s.h5' % model_suffix
        else:
            model_file = 'models/conv_model_%s.hdf5' % model_suffix
        emotion_map_file = 'models/conv_emotion_map_%s.json' % model_suffix
        emotion_map = json.loads(open(resource_filename('EmoPy', emotion_map_file)).read())
        return load_model(resource_filename('EmoPy', model_file)), emotion_map


    def _print_prediction(self, prediction):
        if self.verbose:
            normalized_prediction = [x/sum(prediction) for x in prediction]
            for emotion in self.emotion_map.keys():
                print('%s: %.1f%%' % (emotion,
normalized_prediction[self.emotion_map[emotion]]*100))
            dominant_emotion_index = np.argmax(prediction)
            for emotion in self.emotion_map.keys():
                if dominant_emotion_index == self.emotion_map[emotion]:
                    dominant_emotion = emotion
                    break
            # print('Dominant emotion: %s' % dominant_emotion)
            # print()
        else:
            print('verbose is False')
        return prediction
```

```
target_emotions = ['anger', 'fear', 'surprise', 'calm']
model = FERModel(target_emotions, verbose=False)


prediction = model.predict('nicpic.jpg')


normalized_prediction = [x/sum(prediction) for x in prediction]


result_dict = {}
for emotion in model.emotion_map.keys():
    # print('%s: %.1f%%' % (emotion, 
normalized_prediction[model.emotion_map[emotion]]*100))
    result_dict[emotion] = normalized_prediction[model.emotion_map[emotion]]*100
result_dict
```

Appendix 7.3.1.2 Source code for emotions scoring, partially modified version of the Emopy package (https://github.com/thoughtworksarts/EmoPy), Perez, A. (2017)