The Effect of LinkedIn Profile Pictures on the Probability to Work for Apple

By

Thanakij Wanavit

A Case Study Research submitted in partial fulfillment

of the requirements for the

Degree of Master of Business Administration

Supervisor: Dr. Leslie Klieb

Webster university

Thailand Campus

<Date>

Index

Abstract

In an attempt to create a mathematical model to predict the likelihood of a person working for Apple using LinkedIn profile pictures, the study analyzes the emotions differences between samples from three groups of profile pictures: Apple software engineer in the San Francisco Bay Area, other software engineers in the Bay Area, and US Population average.

Each image is passed through an open-sourced EmoPy algorithm (Perez 2018) which forecasts the extent to which a person feels each emotion in the supported list of trained emotions.

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

When an emotion score is looked at in 1 dimension such as in the case implemented by an F-test, a very weak difference between each group was found which was not sufficient to differentiate between each group, hence a one-dimensional linear regression forecasting model is unlikely to succeed. However, when the output data is looked at in multi-dimension such as the case of the Random Forest method, the predictor model was able to classify the subjects with approximately 62% accuracy compared with 50% base accuracy. The confusion index confirms that the classifier obtains similar score for recall, precision and F1 scores. The success of the Random Forest model infers that there is a real deep relationship which is strong enough to be detected. A well optimized deep neural network with sufficient is probably able to exploit and make a more accurate model than Random Forest since it is designed to capture deeper dimensions.

It is reasonable to conclude that any individual emotion observed by the EmoPy algorithm while taking the profile is not a good indicator of the likelihood of a person working for Apple. However, the combinations of emotions can be used to indicate the likelihood with at least 62% accuracy. It is likely that there are other components of the profile images that has a much larger influence on the likelihood for example skin color, dress, facial features, and hair features.

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 relate to the probability of the person to work as a software engineer for Apple in the San Francisco Bay Area.

1.2 Purpose of the Study

The purpose of the study is to explore the relationship of male persons working 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 and retention 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 male software engineer in the Bay area of San Francisco affect his probability to work for Apple?

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 in most parts of the British Empire (but not England itself) to enslave humans who were visually black. This ended when the English parliament passed the Slavery Abolition Act, (1833) which took effect on 1 August 1834 ("Slavery Abolition Act | History & Impact", 2020). Slavery in the U.S.A. was only abolished in 1865.

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 leads to income discrimination based on appearance. One study by Hamermesh & Biddle (1994,1174-1186) concluded that in the USA and Canada, "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 an 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. The proven influence of such facial factors as beauty leads to the question if other facial factors, like the expression of emotions, also have economic influences. This is the topic of this research.

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 posts (Broniatowski et al., 2014). In addition to physical disease detection, there has been interesting research on mental disease including Kotikalapudi et al., (2012) where the correlation between web usage pattern and depression was analyzed. 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 mirror 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 outlets have been built for a specific purpose. For instance, LinkedIn is optimized for sharing a professional profile to potential employers and colleagues, GitHub is built for sharing coding work with other developers or potential employees, Twitter is for sharing opinions with the public, and Stack Overflow is for sharing problems and solutions 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)

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 to 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 et al., 2010) and improving self-esteem (Gonzales & Hancock, 2011; Steinfield et al., 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 be associated with (Grasmuck et al., 2008). Research has shown that popularity-seeking users tend to disclose more information on Facebook (Christofides et al., 2009; Utz et al., 2012), in order to promote their profiles (Utz et al., 2012). Social media profiles are one of the best tools to represent the personality of individuals (Back et al., 2010).

2.3 Exploring embedded data in images

2.3.0 Neural Networks

This thesis uses neural networks in order to extract what kind of emotions a LinkedIn picture displays. A neural network is a kind of machine learning algorithm which consists of layers of algorithms, each algorithm is called a neuron. Each neuron connects to each other to form a network which can be visualized in Figure 2.3.0.1. (Chen, 2020). Neural network algorithms are widely used in image-related algorithms including EmoPy, which is designed to extract the emotion of a person in the picture.

Neural network can be trained in order to create a meaningful output. The act of training means feeding the network with a set of targeted input and targeted output called ground truth. The ground truth outcome is compared to the neural network calculated output. The deviation between the two outputs is used to modify the parameters of functions in each node in order to get the calculated output to be closer to the ground truth value using a method called gradient descent (Kobayashi, 2017).

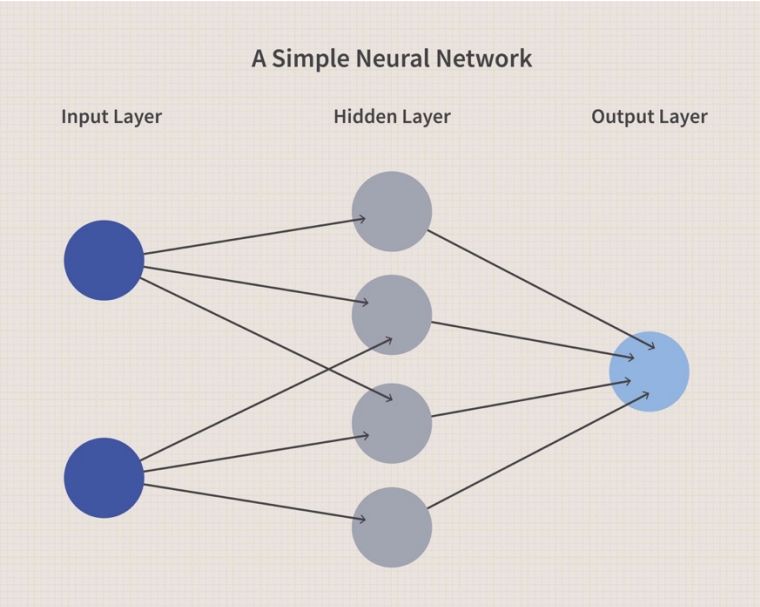
A Convolutional Neural Network (CNN) is a neural network which contains at least one convolutional hidden layer (Figure 2.3.0.1) The difference between Convolutional and other layer is that Convolutional layers utilize a 3-dimensional tensor non-linear function instead of a 2-dimensional matrix used in a linear layer. The difference between different type of layer is visualized in diagram 2.3.3.1.

This chapter gives a list of the steps used in analyzing the emotional content.

1. A neural network published under the name of EmoPy was downloaded from GitHub https://github.com/thoughtworksarts/EmoPy. This is modified as detailed section 2.3.0.1
2. The Modified EmoPy algorithm is trained with FER+ labelled dataset which is discussed in section 2.3.3.1
3. The output of the EmoPy algorithm which gives the score for the degree of each emotion in the output list table 2.3.0.2
4. The output is normalized to that they sum to 100 using the Softmax algorithm shown in equation 2.3.2.1

Figure2.3.0.1

An Example of a neural network (Chen, 2020)



2.3.0.1 The EmoPy Framework with a CNN Model

EmoPy (Perez 2018) is a Python library which uses a deep neural network to predict human emotional expression classifications given images of people's faces. EmoPy supports the FER model which is trained to produce the output in table 2.3.0.2 which is used in this research. This model is available for download at https://github.com/thoughtworksarts/EmoPy/tree/master/EmoPy/models.

The layer details of this network is shown in Table 2.3.0.3. The network was pre-trained on as a YOLO network (section 2.3.1.2) using the ImageNet dataset (section 2.3.1.4) the repurposed using transfer learning (Yang, 2020). The network is then trained on FER+ dataset (section 2.3.3.1) in order to predict human's emotion instead of the Pascal VOC output. (section 2.3.1.3)

Table 2.3.0.2

List of possible emotions output for Emopy Algorithm

	possible emotions
0	anger
1	fear
2	calm
3	sadness
4	happiness
5	surprise
6	disgust

Table 2.3.0.3

Details of the EmoPy CNN model used (EmoPy 2021)

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

Params means the number parameters available for training. Conv2D means a convolutional layer. MaxPool is a transformation layer which connects convolutional or linear layers of different dimensions together. Flatten is a layer that transform between convolutional and linear layer. Dense means a linear layer. Shape means the input and output tensor dimension.

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 (Viola & Jones 2001) is one of the most common ways 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, p. 21)

In order to understand a social profile picture, it would make sense to use an object detection in conjunction with classification algorithms such as the ones by Papageorgiou and Poggio (2000, p. 23). The classification algorithm can discern which mix of emotions are in the discovered facial object.

2.3.1.1 Implementing an object detection system

The most common approach (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 (Laisheng, 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, including this research. Modern algorithms such as deep neural networks (DNNs) have shown superior performance in a range of different applications (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; Sánchez & Perronnin, 2011). The next section discusses a more suitable approach for this research.

2.3.1.2 YOLO REAL TIME OBJECT DETECTION

The 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) which is a type of neural network used in early work for object detection by tracking each individual object, the approach discussed in the previous section (Redmon et al., 2016), 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. The concept of YOLO is used in the EmoPy algorithm.

2.3.1.3 THE 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. Pascal stands for Pattern Analysis, Statistical Modeling and Computational Learning. VOC stands for Visual Object Classes. The dataset includes a 1.9 GB of training/validation and 1.8 GB of testing dataset for 1530 different images of objects and people (Four categories: Vehicles, Household Objects, Animals, and Others. The first three

consisted of different categories of objects, Others consisted of Persons only. In total there were 20 classes). (Everingham et al., 2009) (Pascal2, 2012)

2.3.1.4 IMAGENET DATASET

The ImageNet dataset (image-net, 2012) was published in order to be used in the ILSVRC competition in 2012 (Russakovsky et al., 2015). The goal of the competition was to estimate the content of photographs for the purpose of retrieval and automatic annotation using a subset of the large hand-labeled ImageNet dataset (10,000,000 labeled images depicting 10,000+ object categories) as training. This has become the main public dataset that is used to pre-train neural network before transferring to other datasets by transfer learning (Yang, 2020).

It has been shown that there is a great benefit to use a pre-trained classification model which has performed well on a large dataset such as this for transfer learning. (Yosinski et al., 2012) Transfer learning is a technique of repurposing an existing classification model to classify a different type of data. For example, a dog breed classification algorithm which classifies images of dogs based on breeds can be repurposed to classify species of flowers. The fact that the model has previously performed well on the dogs dataset, means it already has a good understanding of the images and how its features are related.

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 et al., 2017)

While all the methods are valid and are still used in many cases, a neural network allows all of the methods to be performed simultaneously using GPU acceleration (the graphics co-processor unit). This results in the benefits of all the methods without a significant overhead. Google Colab (Google, n.d.), a service from Google to run the code on a virtual machine instance that give free access to a GPU, was used for processing the images in this research using the GPU.

2.3.2.2 Classification with neural network

For each of the four face recognition methods in 2.3.2.1, each of the seven emotional features in each LinkedIn picture is assessed programmatically and given a score which indicates the strength of that emotion. These scores are normalized so that they sum to 100. Collectively, these emotional features should summarize all the useful information considered in the image in this research while reducing the amount of data fed into the final layer of image classifier, which is usually, and also in this research, a fully connected neural network layer with SoftMax activation function (figure 2.3.2.1) which outputs a probability of the classification

result. In Simonyan & Zisserman (2015) 's case, it represents the probability of the object containing each of the 1000 pre-trained categories including dog breeds as required by the ImageNet competition, detailed in section 2.3.1.4. 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 et al., (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.

Figure 2.3.2.1

Softmax activation function (Ji et al., 2018)

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

In this formula, z_l, l=1..k are the input vector, k is the number of classes, (z)i is the output vector.

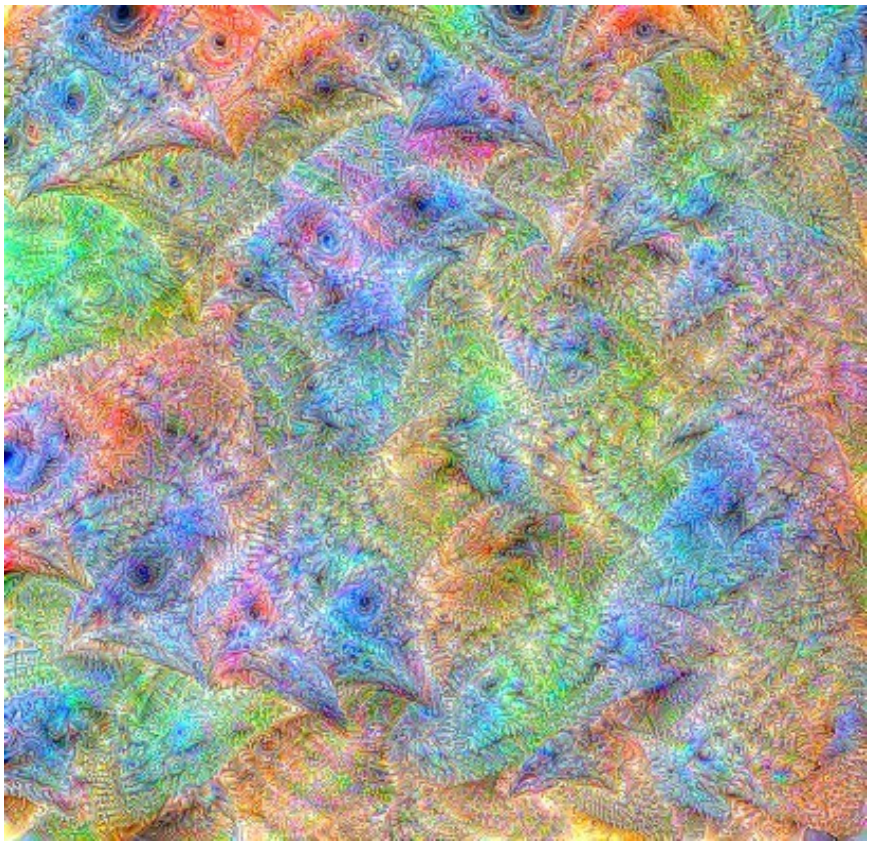
*A good introductory explanation is given in " Softmax Regression, The Softmax Function, Simplified " TowardsDataScience (*Mahmood 2018). T*he output softmax(z)$_I$ is a number between 0 and 1 and can therefore be interpreted as probability.*

2.3.2.3 Visualizing Neural network layers in an image

The method outlined by Graetz (2019) 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, 2019)



This is an example of what a neural network perceives when passed an image of an eye

2.3.3 Combining Object detection and Classification Methods

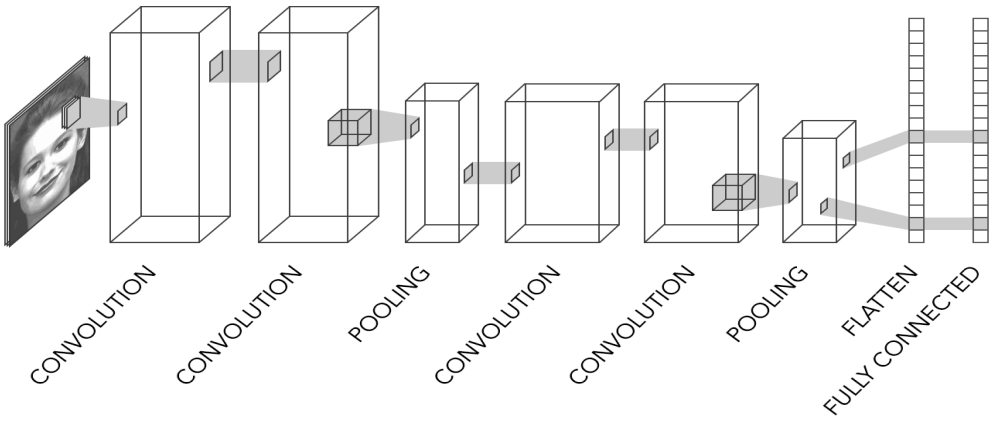
Extraction of the deeper information in an image is possible using a combination of a trained object detection DNN followed by classification DNN. This is demonstrated by an emotion extraction package such as EmoPy (Perez, 2018). This package is used in this work.

Different configurations and variation of Convolutional Neural Networks 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 models 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, 2018)



A convolutional neural network can be used in deep learning for image processing.

2.3.3.1 THE FER+ DATASET

The FER+ annotations provide a set of new labels for the standard Emotion FER dataset. (Kaggle, 2013) This is a set of images of faces. The training set consists of 28,709 examples. The public test set consists of 3,589 examples. Every image is 48 x 48 pixels in grey scale. In FER+, each image has been labeled by 10 crowd-sourced taggers, which provide better quality ground truth for still image emotions 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 et al., 2016). The output of the dataset is one of the 7 emotions in table 3.1.2.1 which is labelled by human.

2.4 Binary Classification Methods

The following binary classification models are used in the research when attempting to forecast the likelihood of an emotion dataset being associated with a person working for Apple.

2.4.0.1 Input

The input to the binary classifier is the output of the EmoPy algorithm which consist of the 7 probability of each emotion in table 3.1.2.1.

2.4.0.2 Output

The output of the binary classifier is the predicted probability of the person working for Apple. This is formatted as a number between 0 and 1. 0 means the person is very unlikely to work for Apple and 1 means the person is highly likely to work for Apple.

2.4.0.3 Rounding and scoring

If the output 0.5 or higher, the output is predicting the person is working for Apple. If the result is correct, then the algorithm gets 1 point for making the correct prediction, and vice versa.

2.4.0.4 Measuring the performance of the classifier

A “Confusion Matrix” (table 2.4.0.4.1) can be used to evaluate the performance and bias of a classification model. This is a set of calculated variables which represents the prediction of the algorithm compared to the ground truth. The “ground truth” is the result observed. From this confusion matrix table, multiple ratios can be calculated in order to judge the quality of the predictions. (Markham 2020) These are listed below. An unbiased classifier should have a balanced accuracy, recall, precision, and f1 score.

Accuracy = (TP+TN) / total samples:

This is the ratio of correct prediction to total describing how often the algorithm is right.

Recall = TP/True samples:

This describes the proportion of correct prediction compared to the positive samples.

Precision = TP/True prediction:

This describes how often the algorithm is correct when it predicts positively

F1 Score = 2/ (Recall-1 + Precision-1): (Sasaki, 2007)

Describes how

Table 2.4.0.4.1

An example of confusion matrix for a binary classification model (Markham 2020)

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

n = the total number of sample, TN = true negative, TP = true positive, FN = false negative, FP = false positive

2.4.1 Random Forest

The Random Forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses averages to improve the predictive accuracy of the model and controls over-fitting. A meta-estimator is an estimator which takes another estimator as a parameter. It therefore combines in a certain way the results of a number of other different methods of making statistical estimates. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement. (Pal, 2005).

2.4.2 Support Vector Machines

Support vector mechanics provides a representation of the training data as points in a (mathematical) space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. (Meyer et al., 2015).

2.4.3 Deep neural network with 1 output

Artificial neural networks as discussed in section 2.3.2.1. use multiple layers of a deep neural network to create a feature set. It is passed through the Softmax (Ji et al., 2018) activation function at the last layer in order to convert the scores of the feature set into a probability.

The Keras-Tensorflow neural network, a library of low-level and high-level API routines (Syed, 2020, p.1) was used in this research.

2.4.4 Logistic regression

Logistic regression is a statistical algorithm for classification using one or more categorical or continuous independent variables. In this case, only categories are used (the different emotions). In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function. (Garg, 2018)

2.4.5 Naïve Bayes

The naive Bayes algorithm is based on the full Bayes' theorem, together with the assumption of independence between every pair of features. Naive Bayes classifiers works well in many real-world situations such as document classification and spam filtering. (Garg 2018) . However, it is not clear in this situation if the emotions are independent.

2.4.5 K-Nearest Neighbors

K-Nearest Neighbors is a classification method by finding distances between a tensor of inputs and the average tensor of the training data for each output groups. The input is classified as whichever output it is closer to. (Altman, 1992)

2.4.5 Perceptron

The Perceptron is an algorithm for a binary classifier which consists of multiple linear threshold function (figure 2.4.5.1) to output a binary forecast. Multiple layers of the perceptron are connected together to form a neural network. (Stephen 1990)

Figure 2.4.5.1

example of a linear threshold function

$$S_i = w_{i,0} + \sum_j w_{i,j} u_j$$

$$u_i = \begin{cases} +1 & \text{if } S_i > 0 \\ -1 & \text{if } S_i < 0 \\ 0 & \text{if } S_i = 0. \end{cases}$$

Source: Researcher, 2021

activations are +1, -1, 0 and ui is the output. wi is the weight(arbitrary constant) , uj is the dependent variable.

2.4.6 XGBoost

XGBoost is a gradient boosting framework which was developed by Chen and Guestrin (2016). This is a kind of distributed deep learning algorithm. The algorithm has been used to win multiple competitions including the Higgs Machine Learning Challange (*Higgs Boson Machine Learning Challenge | Kaggle*, 2015). A gradient boosting is a technique to combine multiple classification models into a single model.

2.5 Role of emotions in emotion expression

2.5.1 Anger expression through facial features

There are complex interactions between the momentary emotions of anger and disgust, contempt, surprise, sadness, and irritation, and the Big Five personality traits on anger expression. This demonstrates the complexity of the predictors of anger behavior in people's daily lives. (Mill et al., 2018, p. 13)

Guo et al., (2015) suggests that anger can be subdivided into “anger-in” and “anger-out”. “Anger-in” is where the user suppresses their anger and do not show it physically, “anger-out” is where they are willing to express it. Although anger in and anger out are clearly related emotion processes, there are also significant differences between the two, by demonstrating that anger in and anger out are influenced by different co-occurring emotions and emotion-personality interactions.

2.6 Databases for storing research data

2.6.1 The S3 Object Store

The S3 object store is an industry leading service from Amazon Web Service which allows user to store and distribute files stored into a folder-like buckets. S3 has multiple advanced feature which are beneficial to the research datastore including, data redundance, version control, accelerated data transfer, high-speed web hosting, and permission control. (Amazon Web Service, n.d.)

The research photos and temporary data logs of this work are stored in S3.

2.6.2 DynamoDB no-SQL data storage

DynamoDB is a database service from Amazon Web Service which allows users to store 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 a continuous built in to prevent accidental data losses. All forecasts data from EmoPy and other Binary algorithms are stored in DynamoDB (Amazon Web Service, n.d.) (Rangel, 2015) (Sivasubramanian, 2012)

2.6.3 Code storage in Colab Jupyter Notebooks

Colab is a free service from Google which allows users to access their virtual machine which contains a GPU access. This is highly valuable for the research since it reduces the time taken to process the data greatly. The code

stored in Colab is automatically backed up on Google Drive and is available to share publicly. (Google, n.d.)

3. Methodology Design

The study aims to collect the data between occupation and face features of male samples from LinkedIn in the region of San Francisco with the company they work for. Computer setup is listed in section 3.3.3

3.0 Summary of research methods

The following steps are done in the research

1. Profile picture and occupation extraction from LinkedIn (section 3.1.1)
2. Prediction of emotional probability with EmoPy using the profile picture (section 3.1.2)
3. Analysis of the data relationship between EmoPy output (from step 2) and the occupation (working for Apple) data Using Emotional probability from section 3.1.2. Details are discussed in section 3.4
    - using ANOVA (section 3.4.1)
    - using binary classification algorithm (section 3.4.2)

3.0.1 Sample Selection

samples are divided into 3 groups:

1. Apple Employees

This is male Apple employees in San Francisco Bay Area with computer engineer title

2. Non-Apple Employees

This is male non-Apple employees in San Francisco Bay Area with computer engineer title

3. control group

This is randomly selected people (males only) in the US

sampling details is discussed in a greater detail in section 3.2

3.1 METHODS

3.1.1 OBTAIN THE DATA FROM LINKEDIN

3.1.1.1 TOOL SETUP

Browser: Google Chrome (Google, n.d.)

WebDriver: ChromeDriver 89.0.4389.23 (Chromium, n.d.)

Python API library: Selenium 3.141.0 (Selenium HQ, 2019)

Programming Kernel: Python 3.6(Python Software Foundation, 2016)

Virtual machine: specification in section 3.3.3

Python script: code in Appendix D

3.1.1.2 AUTOMATION

Although the steps in section 3.1.1.3 could be taken by a human, was done using an automation tools stated in section 3.1.1.1. This is to save costs and minimize human bias. The website LinkedIn cannot differentiate between humans and the automation tool used.

3.1.1.3 STEPS FOR OBTAINING IMAGES FROM LINKEDIN

Browse to www.linkedin.com and click the "search for people" tab. Select the following filter "company: Apple, location: San Francisco Bay Area, occupation: software". The list of people will appear in the search results (figure 3.1.1.3.1). For every second profile in the result list, the profile picture, occupation, and company the person works for is recorded (figure. 3.1.1.3.2). A video demo of the task can be viewed at appendix E.

Figure 3.1.1.3.1

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

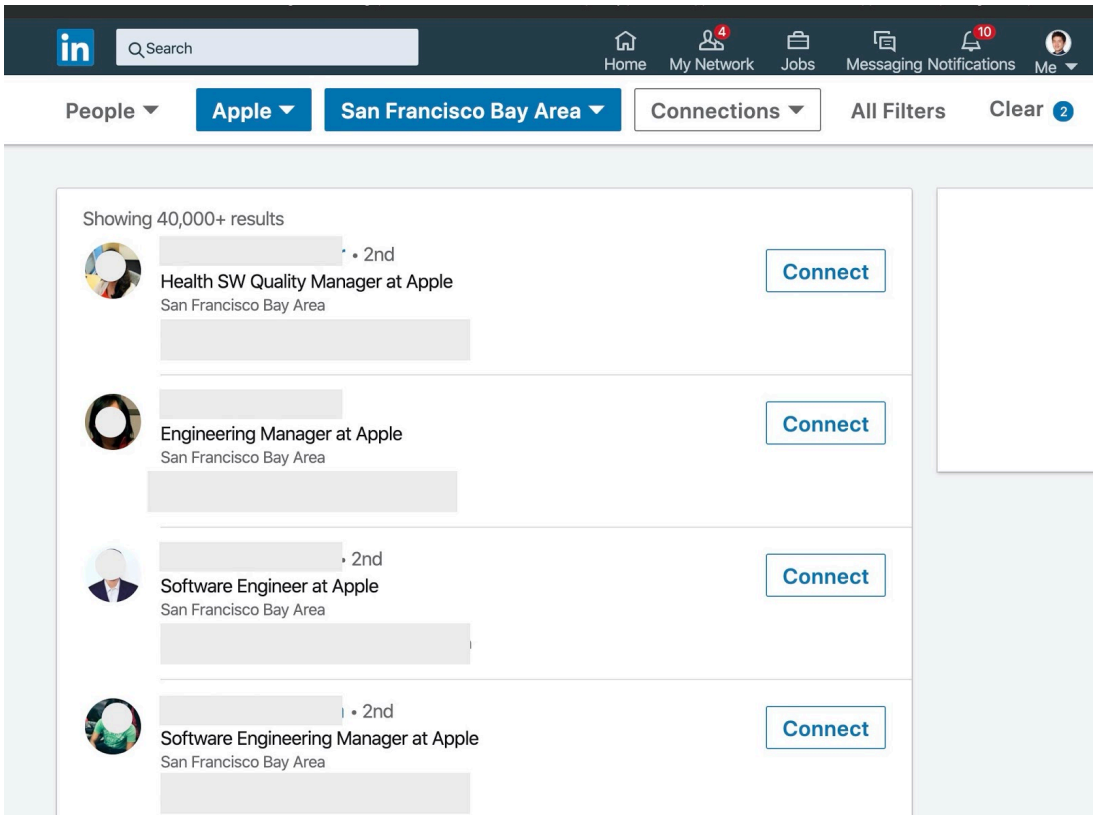
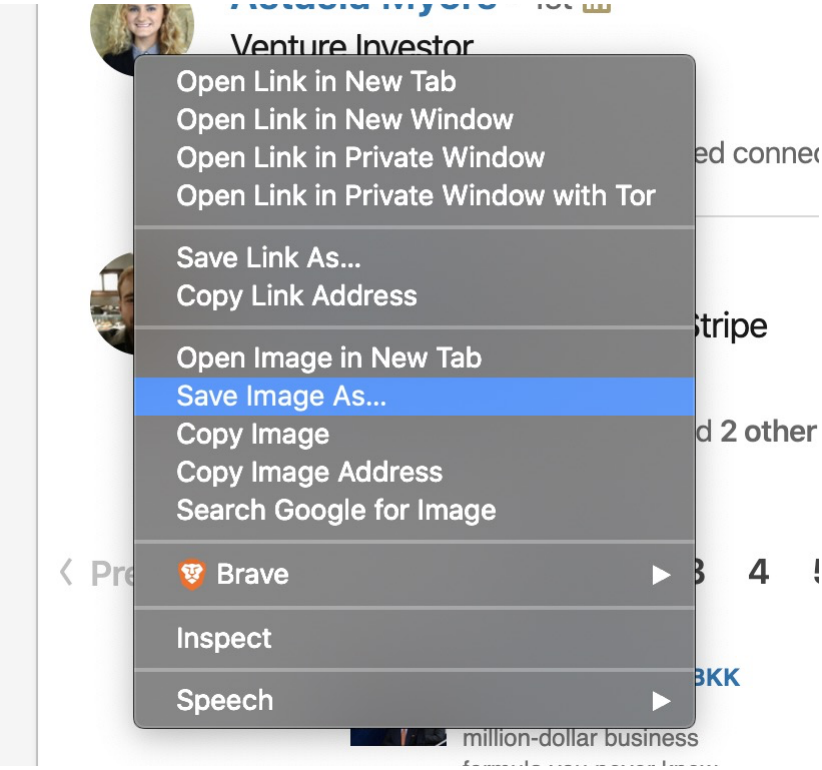


Figure 3.1.1.3.2

How to save an image from LinkedIn



Save image from profile picture within the LinkedIn search page (www.linkedin.com, 10 Feb 2020)

3.1.2 Extract emotional features from each face

The Python package management tool "pip" (Pip, 2021) was used to install the EmoPy package and its prerequisites (section 2.3.0.1). After that, an EmoPy model which is trained on the FER+ (section 2.3.3.1) dataset was installed. This is publicly available for download from EmoPy repository. (ThoughtWorksArts, 2018) (Barsoum, Zhang, Ferrer & Zhang ,2016)

Then the script in Appendix B was used to obtain the probability of each of the 6 the emotion encapsulated within the image. (section 2.3.0.1) The possible emotions are listed in table 3.1.2.1

Table 3.1.2.1

list of emotions of interest

	emotions
1	calm
2	anger
3	happiness
4	surprise
5	disgust
6	fear
7	sadness

Notes:

Scores are recorded for each picture

Source code for the task can be found in Appendix B

A video demo of the following task can be viewed at appendix D.

3.2 SAMPLING

Systematic sampling uses 5 separate robots with an account in Oregon to collect 100 pictures systematically every 2 profiles discovered. This is to minimize the effect of the LinkedIn ranking algorithm that adjusts the served profiles to the profile of the requester, and which is not public.

- Samples in the "Apple" group have the following filter
    - occupation: computer-science
    - area: San Francisco Bay Area
    - employed by company: Apple
    - sex: male
- Samples in the "non-Apple" group have the following filter
    - computer-science
    - San Francisco Bay Area
    - not working for Apple
    - sex: male
- Control group has the following filter
    - "is in United States"
    - sex: male
- Number of samples
    - "Apple": 851
    - "non-Apple": 416

      "control": 517

Note: The number of samples that were collected was done on a best effort basis due to the LinkedIn and computing limitations and does not represent the optimum number of samples.

3.3 Instrumentation

3.3.1 Automation package

Setup for automation is discussed in 3.1.1.1. The automation package aims to imitate and replicate as far as possible the action of human manual data collection with a Python script.

3.3.2 Objects collection

Data collected has been uploaded to an s3 file storage bucket (section 2.6.1) for storage and the URI was stored in a DynamoDB database(section2.6.2).

3.3.3 Computer Setup for the research (Virtual Machine)

(https://aws.amazon.com/ec2/pricing/on-demand/)

OS: Ubuntu18.04

CPU: aws EC2 t4g.medium

RAM: 2GB

HARDDRIVE: 50 GB

REGION: Oregon (us-west-1)

(Amazon EC2, n.d.)

3.4 Data Analysis

3.4.1 Data cleansing and ANOVA analysis

Score on each emotion listed in Table 3.1.2.1 were obtained and put into a single database

The probability for each emotional feature in each group (Apple or others) was analyzed for standard deviation, covariance, and average. Due to the limitations regarding the set-up of some algorithms including the Random Forest, the samples trained and tested needs to have a balanced output. Therefore, an equal number of samples for from Apple and non-Apple was used.

3.4.2 Preparation of data for the multidimensional forecasting model

Due to the balanced sample requirement for the K-Nearest Neighbors algorithm, the sample set is kept balanced by removing some Apple samples so that the total number of samples fed to the model in each run is 416 for either group. For each run of the experiment for error analysis of the Random Forest model (section 3.4.4), this removal of Apple samples is done randomly independent of previous runs.

The samples were divided randomly into training:testing sets with the ratio 80:20. 80% of the data is used to train the binary classification models (section 2.4) and 20% are used to measure the performance of the algorithm according to section 2.4.0.4. There are 416 samples from Apple and non-Apple group totaling 832 sample. This is divided into 666 training samples and 166 testing samples.

3.4.3 Running tests using multidimensional models

Following mathematical models are used in this analysis

- Naive estimate (baseline)
- Logistic Regression (Duncan, DB 1967)
- K-Nearest Neighbours (Naomi S. 1992)
- Support Vector machines (Pupale, 2019)
- Perceptrons (Rosenblatt, 1958)
- XGBoost (Mitchell & Frank, 2017).
- Keras-Tensorflow neural network (Syed, 2020, p. 1) without hyperparameter optimization.

Each model was trained using the code under appendix B using the scikitlearn, XGBoost, and Keras API as commented in appendix B. The models are trained using 80% of the samples then made to forecast whether the person with given emotion is working of Apple or not on the remaining 20% of the samples.

The full code with a runnable Jupyter-notebook instance is made available at the link in Appendix D

The most successful algorithm (Random Forest) was analyzed for the degree of usage of each emotion in the calculation of the final output. Although it has been stated that XGBoost outperforms the Random Forest algorithm, that was not the case in this experiment.

The difference between a single dimensional model (e.g., logistic regression) and a deep model (eg. Random Forest) gives us the information about the depth of relationship. Depth of relationship is the extent of combination of multiple inputs required to generate the output

3.4.4 Error analysis of the Random Forest model

Since Random Forest is the best performing model of the study, an error analysis was done on the model. The model was recreated by shuffling the dataset and randomly splitting into train-test set as in section 3.4.2. The model is then trained and made to forecast as in section 3.4.3. This process is repeated 1000 times and results and plotted and analyzed using z-test to see whether the prediction is significantly higher than the 50% baseline that results from equal-size sample sets.

3.4.5 Specific emotional contribution

It is possible to analyze which emotion is the most important contribution factor to the prediction of Random Forest. This is done according to the Analysis of Variance method. (Zwanenburg et al., 2011, p. 566)

3.5 Control group

As a control group, a sample of images from LinkedIn was used where the only filter was that they were male and living in the USA.

This is useful for determining whether the general programmers in San Francisco Bay Area are significantly different from the general population in term of emotions.

Picture samples with the filter were collected using the method specified in section 3.2.

The control group vs Apple employee group comparison is made to see whether the result of Apple vs Non-Apple in San Francisco Bay Area can be generalized to the general population of USA.

3.6 Random forest model for multidimensional analysis

The Random Forest model is used to analyze 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 ground truth output, the output from the algorithm is normalized using the Softmax function to give the result in a range between 0 and 1 with 0 representing a non-Apple employee and 1 representing 100% confidence that the sample is an Apple employee. The result is then rounded to give a prediction.

If the prediction is made correctly, the score of 1 is given, and 0 otherwise. This score is used to rate the algorithm comparing to baseline which is 50%. This is because of the balanced data preparation of samples according to the section 3.4.2.

4. Significance of the study

The study will allow companies to develop and improve their tools for better decisions based on how they select candidates. Emotional biases can be considered when accepting candidates.

For candidates who would like to apply for jobs, this understanding may give an insight to the chance of being accepted and retained at a specific job in a company. This would help them choose a suitable job

5.Analysis and findings

5.1 Results

5.1.1 Overall Results

The data collected are arranged into an SQL table as shown in 5.1.1.1. 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 Figure 6.1.1.2 shows that the profile picture images are showing the same pattern trend for all three categories. However, this does not mean that there are no differences. More details are explored when comparing a higher dimension spaces in the next section.

5.1.1.1 General Statistics

The number of samples analyzed as shown in table 5.1.1.4 shows that have collected 851 samples of Apple, 416 samples of other companies in the Bay Area and 517 from the general US population. The number of subjects fed to the binary classification model is reduced to 416 for Apple and 416 for non-Apple employees to keep each binary group equal. These numbers are due to limited time and resources available for data collection. The researcher tried to collect as many samples as possible in the limited time they had before LinkedIn blocks the access. More data can be collected given more time and resources. table 5.1.1.5 and table 5.1.1.6 shows standard deviation and mean of the scores received for each sample. table

5.1.1.11 shows the t-score for the difference between sample in Apple and non-Apple employees for each emotion.

Table 5.1.1.4

Summary of number of samples collected for each group

		count
isAppleEmployee	**isControl**	
False	**False**	416.0
	True	517.0
True	**False**	851.0

Table 5.1.1.5

Summary of mean of sample scores (%probability) collected for each group

		surprise	calm	happiness	sadness	disgust	anger	fear
isAppleEmployee	**isControl**							
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

Table 5.1.1.6

Summary of standard deviation of sample score collected for each group

		surprise	calm	happiness	sadness	disgust	anger	fear
isAppleEmployee	**isControl**							
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

Table 5.1.1.7

Summary of coefficient of variation (standard deviation/mean) of samples collected for each group

isAppleEmployee	isControl	surprise	calm	happiness	sadness	disgust	anger	fear
False	False	0.392628	0.080765	0.355773	1.001008	0.786994	0.342245	0.243889
	True	0.409789	0.082234	0.379832	0.951603	0.824263	0.349830	0.251345
True	False	0.403428	0.083566	0.393576	1.008626	0.814143	0.347516	0.253562

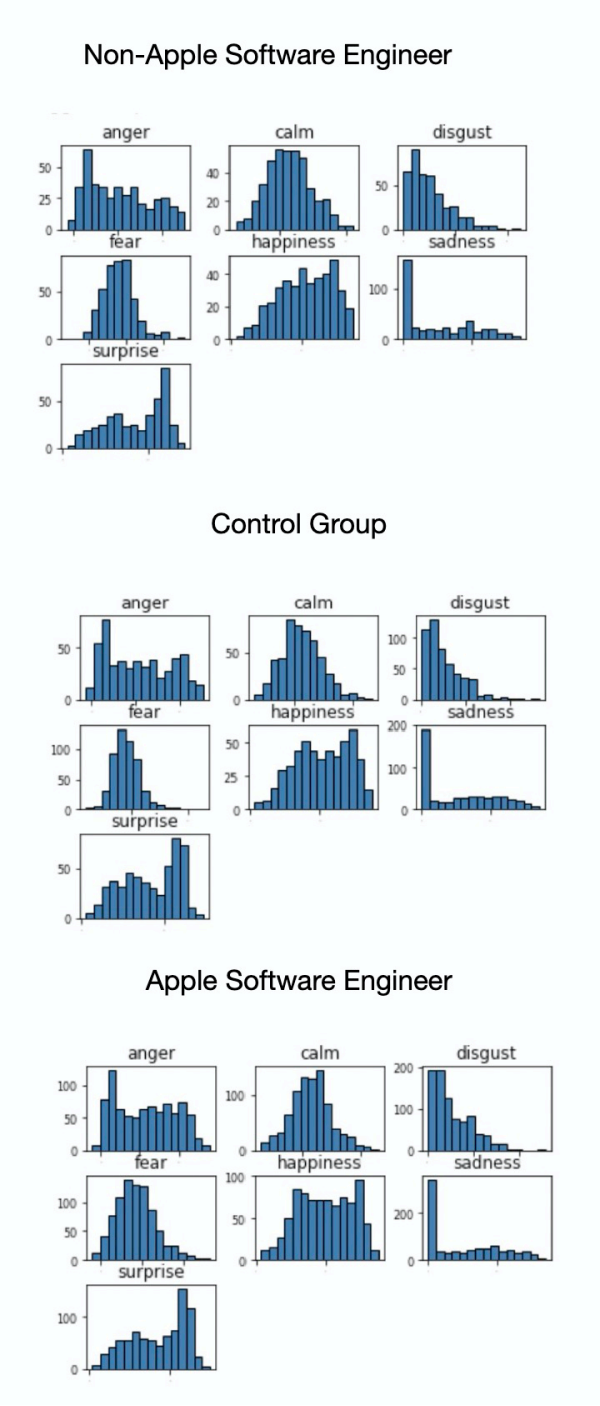
Table 5.1.1.8

Individual score of the first 5 candidates in the database

	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

Figure 5.1.1.9

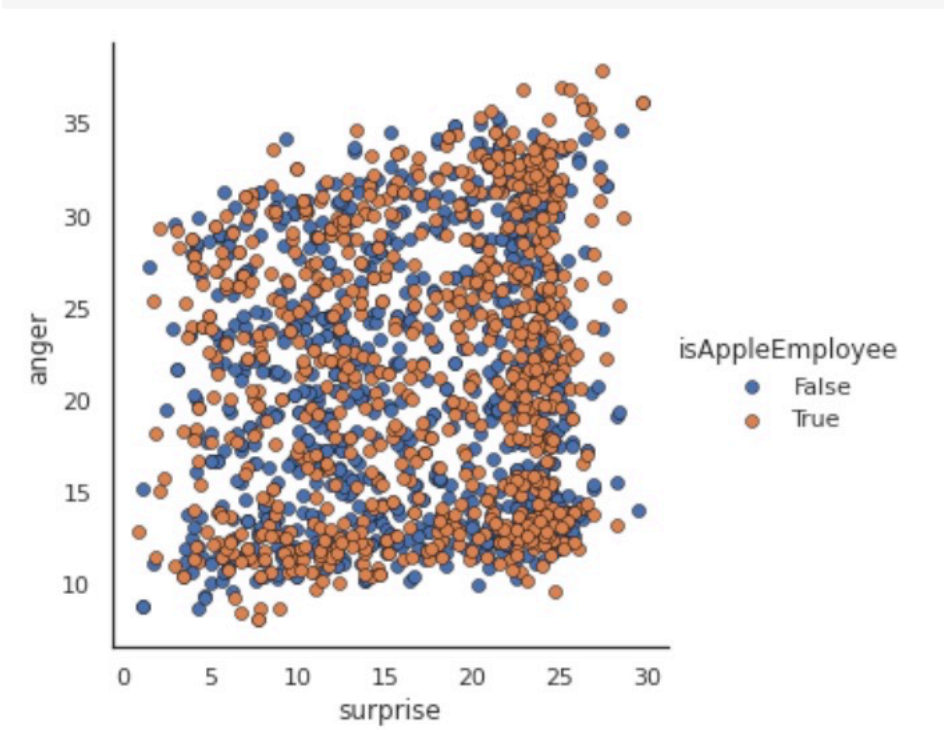
Summary histogram over all Emotions for each group of samples



The histograms shows that each group of participants exhibit similar but different pattern in each emotion's probability distribution. Source: Researcher, 2021

Figure 5.1.1.10

Scatter plot between anger and surprise comparing Apple employee and general population



The table plot shows that there is no obvious pattern between the plotted emotions and whether they work for Apple. Source: Researcher, 2021

Table 5.1.1.11

T-Score for each emotion comparing Apple and Non-Apple employees

	surprise	calm	happiness	sadness	disgust	anger	fear
t-score	-0.492342	-0.849648	2.814996	0.934658	-0.737279	-2.984136	-1.794707

5.2 Comparisons between software engineers who work for Apple and other companies in the bay area

5.2.1 General results for the groups

Figure 5.2.1 shows a table of comparison between each category in software engineer at Apple and not at Apple. There is a small difference in the mean between happiness disgust and anger. The t-test in Table 5.1.1.11 shows that happiness, anger, and fear have a mean that is statistically different with a t-score outside the range ± 1.282 which is 90% confidence interval critical value (NIST/SEMATECH, 2012). Table 5.1.1.7 has shown that sadness and disgust have the highest coefficient of variation which means that the value varies the most between person to person and may be less useful as a predictor.

5.2.2 Comparison in 3 dimensions

Figure 5.2.2 and 5.2.3 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 shows that these groups may share much of the same emotion when presenting their profile picture in general on LinkedIn.

Figure 5.2.1

Scatter diagrams for each category of emotion for Apple versus others for software engineers

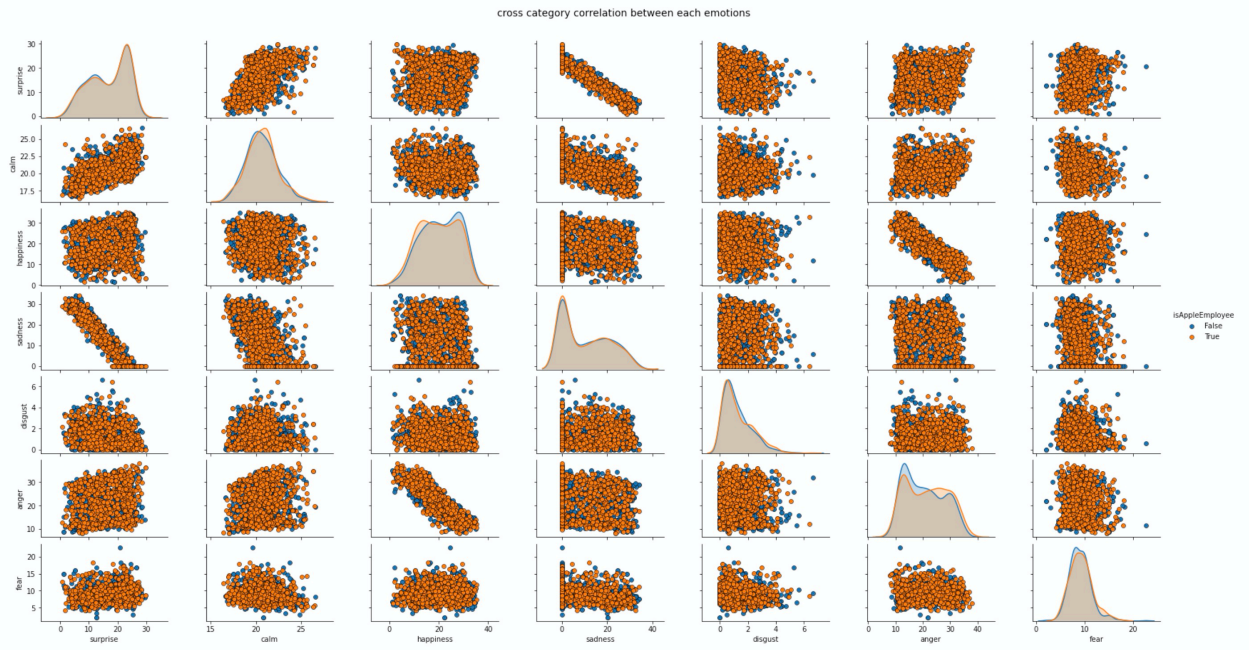


Figure 5.2.2

3D plot for emotions [anger, happiness, and disgust] for the Bay area samples. Source: Researcher, 2021

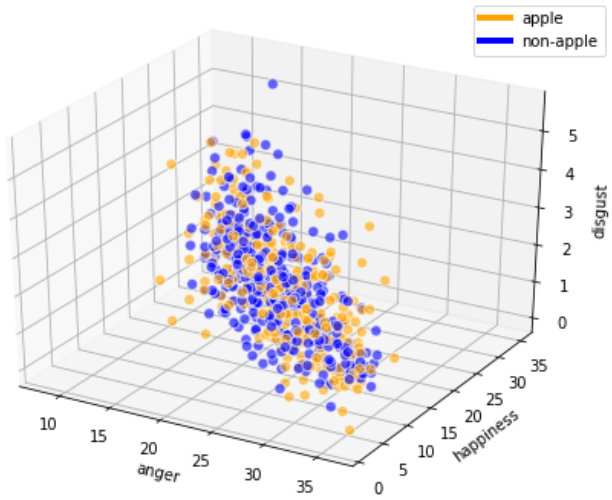
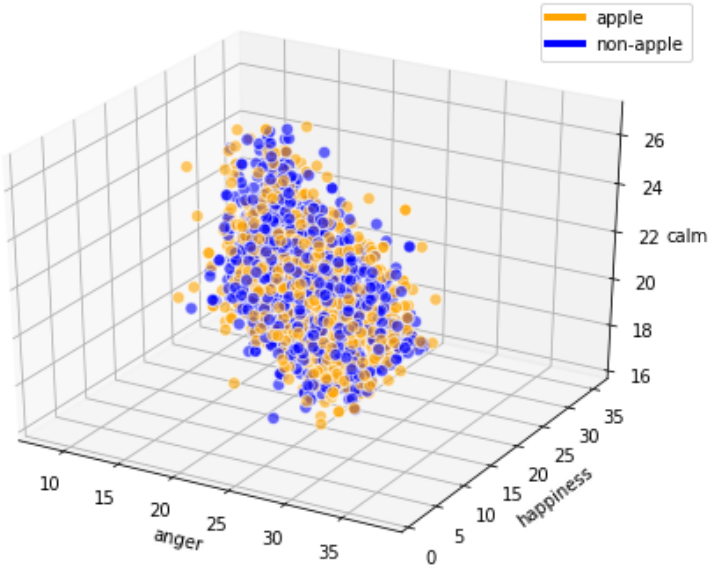


Figure 5.2.3

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

Researcher, 2021



5.2.3 Comparison Multiple Dimensions

Machine learning algorithm can be used to determine a deeper relationship. Data is split randomly 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 samples used is 416 per group after data preparation in section 3.4.1. This gives the baseline performance using Naïve forecast at 50%

5.2.3.1 Machine Learning models result comparison

Figure 5.2.3.1 shows that most models perform better than the baseline of 50%. The Random Forest model is able to forecast with an average of 59.2 ± 2.5% (table 5.2.4.1). This is slightly lower than 63.0% when forecasting between Apple and control group in section 5.3. The accuracy is significantly more than 50% after the errors is taken into account (z test in section 5.2.4.2.). The confusion matrix shows that these forecasts are balanced in term of precision, recall and f1-score. We are able to conclude that there is a deep relationship between the two groups. It is possible that a more optimized model with more data and optimal hyperparameter can outperform the model used in this experiment. Definition of confusion matrix terms and baselines is described in Appendix C.

5.2.3.2 Importance of each emotion

The Random Forest model is the best model in the experiment therefore, each feature is plotted for their contribution in the forecasting model. Figure 5.2.3.2 shows that disgust and fear emotions are the most important when trying to determine the group of samples.

Figure 5.2.3.1

Prediction score for each machine learning model. Source: Researcher, 2021

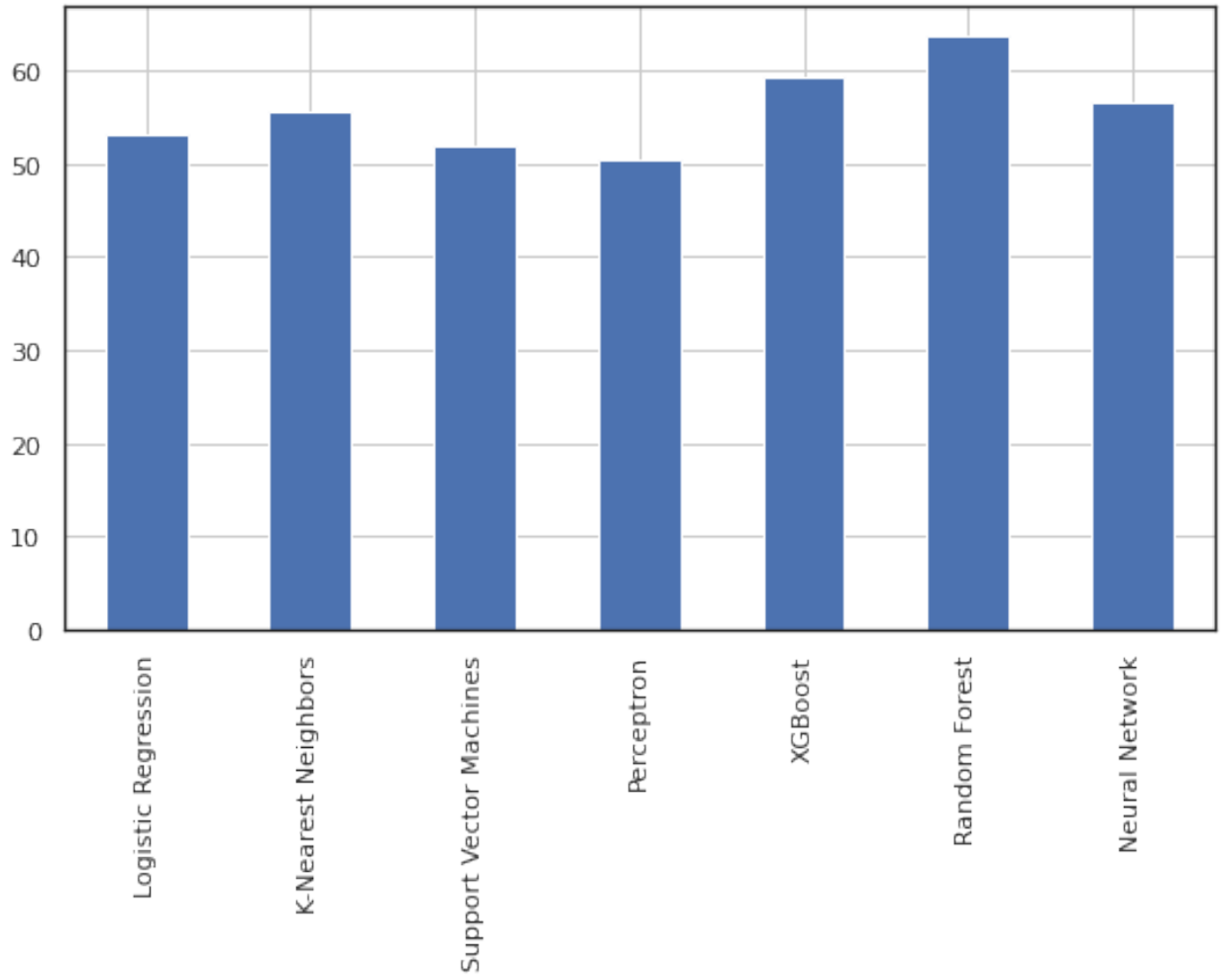
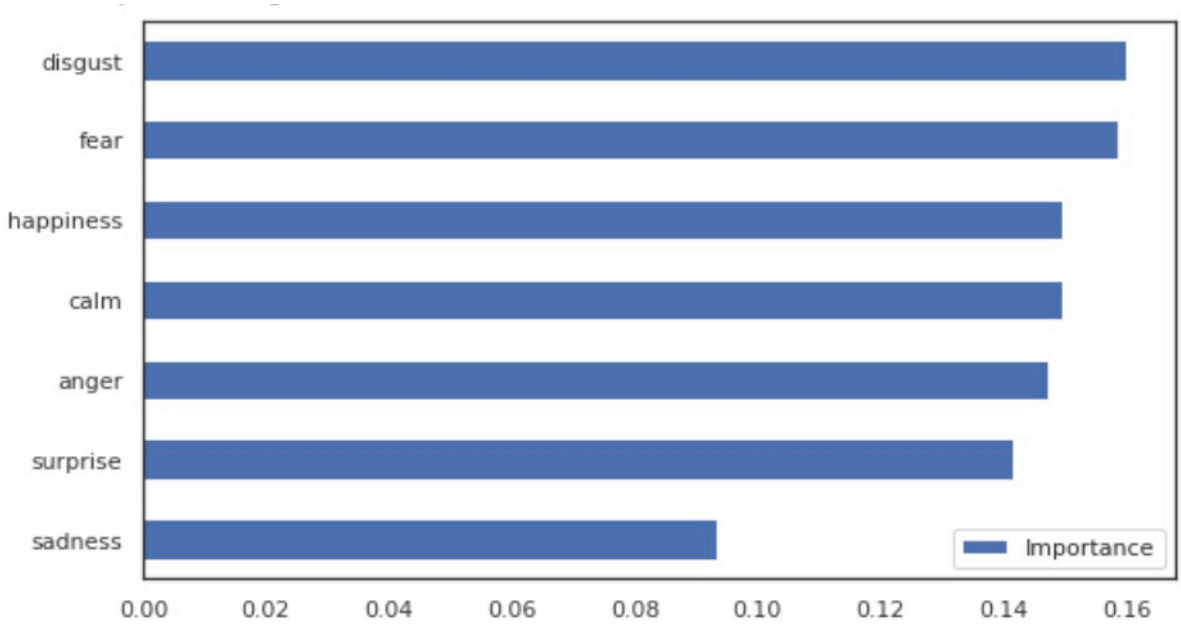


Figure 5.2.3.2

The importance of each emotion based on Random Forest model used in Fig 5.2.3.1. Source: Researcher, 2021



5.2.3.3 Confusion index and scoring for the best algorithm

The best algorithm for forecasting in the study is the Random Forest algorithm. The confusion index (fig.6.2.3.3) shows that the algorithm receives similar score for precision, recall, f1-score. Out of 166 testing samples (see section 3.4.2. for more details), 83 from each of Apple and non-Apple group, Random Forest forecasted 50 true positives, 53 true negatives, 30 false negative, and 33 false positives. This means the algorithm is performing in a non-biased way and the relationship is statistically significant (section 5.2.4). Non-biased meaning it is predicting a balanced number of correct positives and negatives.

Figure 5.2.3.3

Confusion matrix and the performance indexes for the Random Forest algorithm

```
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
```

5.2.4 Confidence of accuracy score of Random Forest

After Repeating the Random Forest analysis 1000 times, the result is analyzed for its general statistics and z-test

5.2.4.1 General Statistics

Table 5.2.4.1 shows that the mean forecast is 59.19 over the 1000 samples with the standard deviation of 2.46. Every single trial has an accuracy score higher than 51% with maximum score of 65.4%. It is clear that the forecast has consistently outperform the naïve method (score 50%) and there is a deep relationship. The z test is shown in section 5.2.4.2

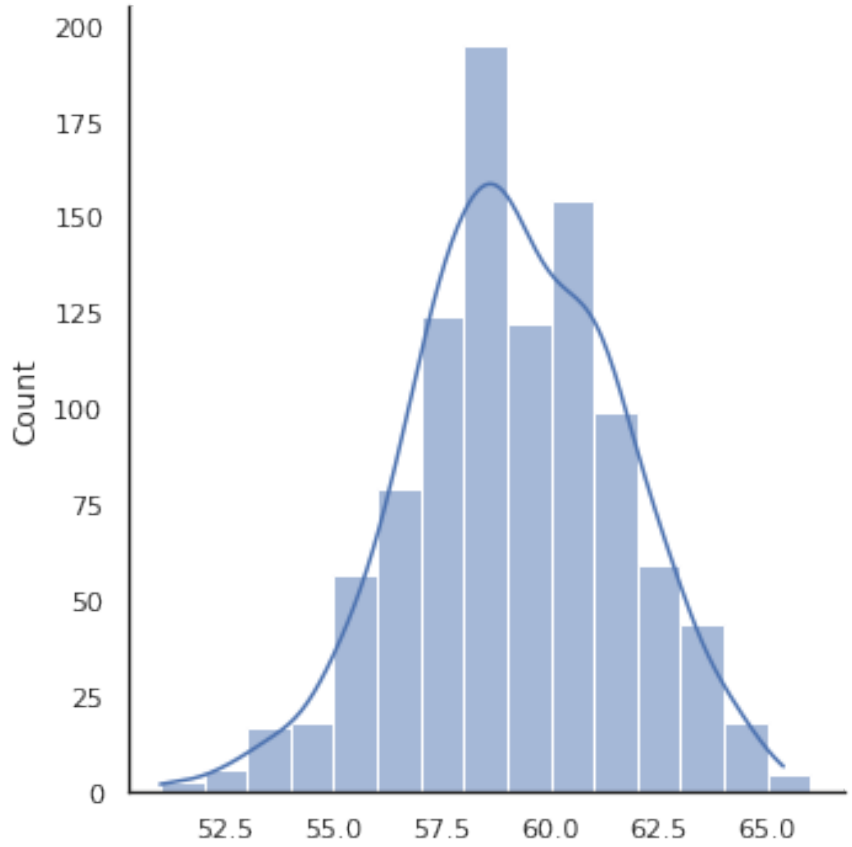
Table 5.2.4.1

General Statistics of the Random Forest forecasts

count	1000.000000
mean	59.190800
std	2.463068
min	51.030000
25%	57.480000
50%	59.240000
75%	61.000000
max	65.400000

Figure 5.2.4.1.2

Countplot of Random Forest forecasts



The count plot shows a bell shaped chart which justifies Gaussian Distribution approximation used in the z-test. Source: Researcher, 2021

5.2.4.2 Z-test of Random Forest forecasts

The 1 tail z-test is done with the null hypothesis that the naïve forecast gets the same or higher score than the Random Forest forecast. The alternative hypothesis is that the Random Forest forecasts has a higher score than the naïve forecast.

Figure 5.2.4.1.2 shows that the shape of the data can be approximated using Gaussian Distribution. This means that z-test can be justified as a good estimator of the results.

The z score is 3.73, which means that the probability that Random Forest has a performance of 50% or worse is 9.52e-05. This is much lower than the alpha which is 5e-2 therefore the null hypothesis can safely be rejected.

Equation 5.2.4.2

z-test equation used in the analysis

$$Z = \frac{(\bar{X} - \mu_0)}{s}$$

5.3 Comparison between Apple software engineer and the 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 62.0±2.5% accuracy compared with a baseline of 50% This signifies that there is a significant deep relationship between Apple and control group employees when data is looked at in 7 dimensions. Fig 5.3.6 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.

Figure 5.3.1

Cross correlation plot between Apple employees and American average person. Source: Researcher, 2021

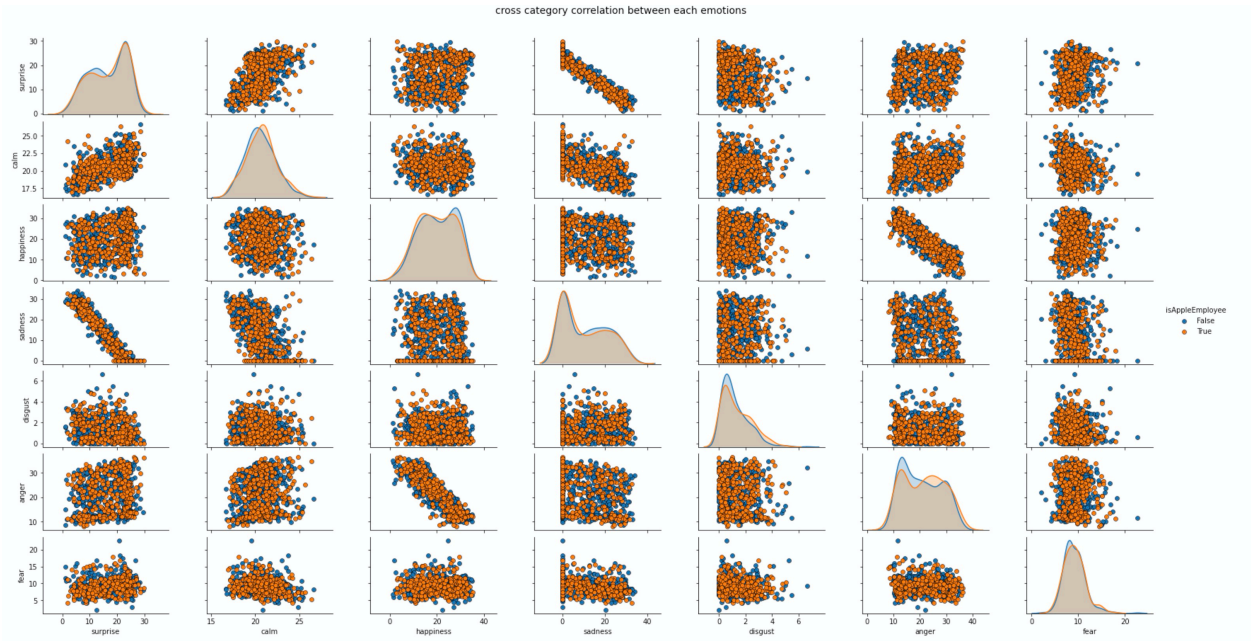


Figure 5.3.2

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	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

Figure 5.3.3

Mean emotion score (as % probability) 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	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

Figure 5.3.6

The contribution of each factor to the Random Forest model. Source: Researcher, 2021

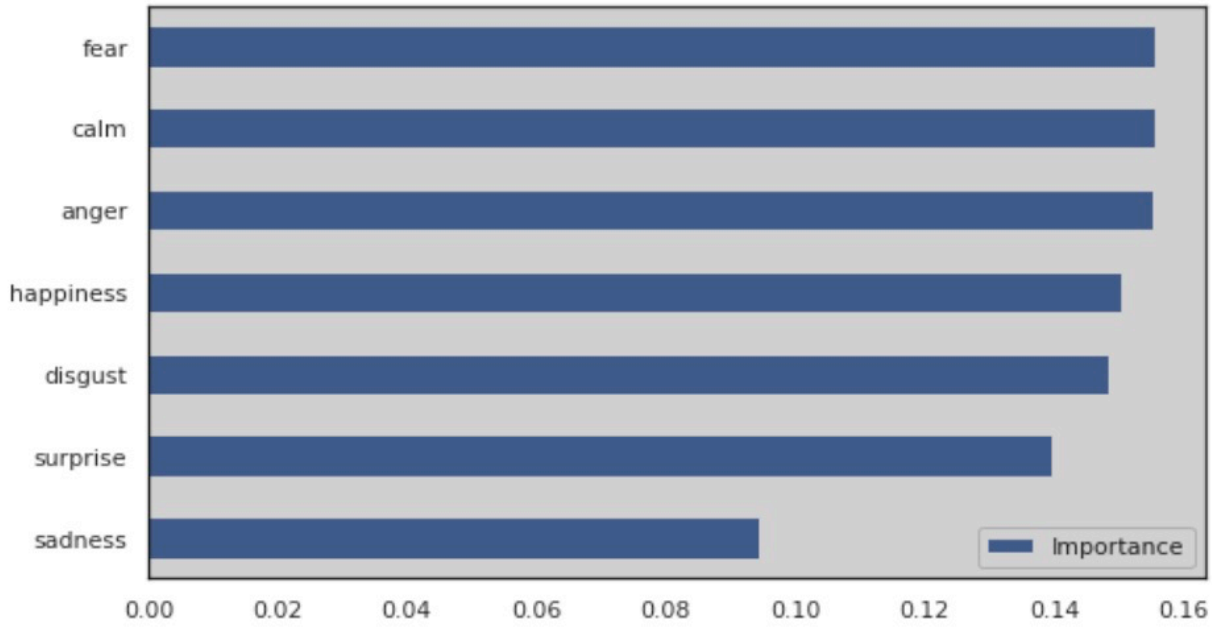
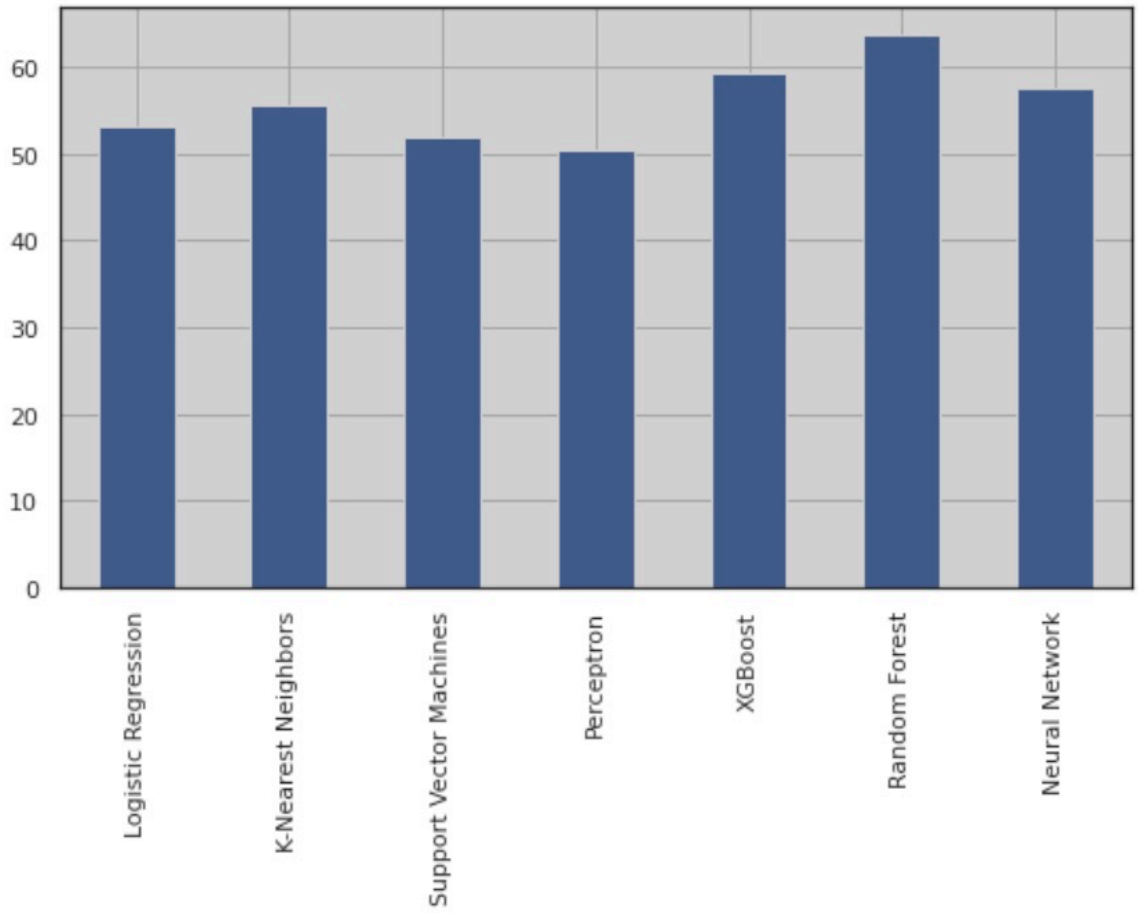


Figure 5.3.7

Forecast comparison by various machine learning models. More than 50% indicates relationship between Y and X data. Source: Researcher, 2021



5.4 Limitations and Delimitations

5.4.1 Limitations

A Company like Apple hires and retain personnel for many different functions from cleaning, reception to business managers. The criteria for the suitability for each position could be vastly different. This research limits itself to Computer Science employees.

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

Nationality, race, education background, gender and working location may affect the profile picture and the chance to work for Apple.

5.4.2 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 with different hiring criteria are not included in the study.

Pictures were 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 are assumed to share many characteristics and behaviors due to the vicinity.

To reduce bias due to LinkedIn's recommendation algorithm, the robot logs in from multiple locations and create 5 LinkedIn profiles and logs in using different IPs during the study.

The area of the experiment was set to San Francisco Bay Area in order to limit the diversity of working location. The job of Software engineer should somewhat limit the diversity of the education background. Majority of the population working for Apple is US citizen although the ethnicity is very diverse with 25% Asian, 25% Latino, and 40% White.(bayareaequityatlas. n.d.)

6. Conclusion

6.1 Conclusion Statement

The current results suggest that there is a relationship between the emotional features showing in the LinkedIn pictures that software engineers show on their profile pages and if they work for Apple or not. The relationship seems not very strong but persistent. There is a deep but unclear relationship between facial expression on the profile page and the probability that a person is working for Apple. Although the experiment has not shown a clear correlation between the likelihood of working for Apple and each individual emotion, the Random Forest in combination with the EmoPy algorithm suggests that there is a statistically significant relationship between the image and the likelihood of the person working for Apple.

The fact that no single emotion is as good predictor as the Random Forest model suggests that the relation may be due to other aspects of facial features which may not have a direct link to emotion. This study did not collect demographic data. It is possible that there were differences between the groups that influenced the emotional features that seem to affect the outcome. Age could be such a factor, race too (it is known that there are racial differences in the way face-reading algorithms perform). More studies with a deep neural network or using algorithms that are focused on other facial features may help to understand which facial feature leads directly to the likelihood of working for Apple.

More studies should be done to explore deeper into the nature of the relationship and how companies will be able to use this information to

minimize bias and discrimination against an applicant with a specific profile picture. Other companies and location can be analyzed using similar concept to make a more generalized conclusion.

6.2 Summary of findings

Although individually, emotions in the profile pictures are not a good predictor whether a person is likely to be an Apple employee, when looked at in 7 dimensions, a standard Random Forest model is able to predict with the average of 59 ± 2.5% accuracy compared with a baseline of 50% whether a person is working for Apple. The algorithms in this research can still be improved using an optimized version of Random Forest or an optimized deep neural network models with different set of hyperparameters and more data for training.

7. References

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

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, *46*(3), 175–185. https://doi.org/10.1080/00031305.1992.10475879

Amazon Web Service. (n.d.). *Amazon EC2*, Inc. Retrieved September 7, 2020, from https://aws.amazon.com/ec2/

Amazon Web Services, Inc. (n.d.). *Introduction to Amazon DynamoDB (1:01)*. Retrieved November 22, 2020, from https://aws.amazon.com/dynamodb/

Back, M. D., Stopfer, J. M., Vazire, S., Gaddis, S., Schmukle, S. C., Egloff, B., & Gosling, S. D. (2010*). Facebook Profiles Reflect Actual Personality, Not Self-Idealization. Psychological Science,* 21(3), 372–374. https://doi.org/10.1177/0956797609360756

Barsoum, E., Zhang, C., Ferrer, C. C., & Zhang, Z. (2016). *Training deep networks for facial expression recognition with crowd-sourced label*

distribution. In Proceedings of the 18th ACM International Conference on Multimodal Interaction (p. 279-283). https://doi.org/10.1145/2993148.2993165

Bayareaequityatlas. (n.d.). Race/ethnicity | Bay Area Equity Atlas. Bayareaequityatlas.Org. Retrieved April 5, 2021, from https://bayareaequityatlas.org/indicators/race-ethnicity#/

Brinker, T. J., Hekler, A., Enk, A. H., Klode, J., Hauschild, A., Berking, C., ... & Schrüfer, P. (2019). Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. *European Journal of Cancer*, *113*, 47-54. https://doi.org/10.1016/j.ejca.2019.04.001

Back, M. D., Stopfer, J. M., Vazire, S., Gaddis, S., Schmukle, S. C., Egloff, B., & Gosling, S. D. (2010). Facebook Profiles Reflect Actual Personality, Not Self-Idealization. *Psychological Science*, *21*(3), 372–374. https://doi.org/10.1177/0956797609360756

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

Broniatowski, D. A., Paul, M. J., & Dredze, M. (2014). *Twitter: Big data opportunities. Science*, *345*(6193), 148. https://doi.org/10.1126/science.345.6193.148-a

Cash, T. F., Gillen, B., & Burns, D. S. (1977). *Sexism and beautyism in personnel consultant decision making*. *Journal of applied psychology*, *62*(3), 301.

Chen, T., & Guestrin, C. (2016). *XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 1. https://doi.org/10.1145/2939672.2939785

Chen, J. (2020, December 23). *Neural Network Definition*. Investopedia. https://www.investopedia.com/terms/n/neuralnetwork.asp

Christofides, E., Muise, A., & Desmarais, S. (2009). *Information Disclosure and Control on Facebook: Are They Two Sides of the Same Coin or Two Different Processes? CyberPsychology & Behavior,* 12(3), 341–345. https://doi.org/10.1089/cpb.2008.0226

Chromium. (n.d.). *ChromeDriver - WebDriver for Chrome*. https://Chromedriver.Storage.Googleapis.Com/Index.Html?Path=89.0.4389.23/. Retrieved February 4, 2021, from https://chromedriver.chromium.org/

Deng, H. (2012). *A Novel UWB Filters design Based on Hybrid Neural Network. Physics Procedia*, *24*, 743–748. https://doi.org/10.1016/j.phpro.2012.02.110

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

Drogosz, L. M., & Levy, P. E. (1996). *Another look at the effects of appearance, gender, and job type on performance-based decisions. Psychology of Women Quarterly*, *20*(3), 437-445.

Emopy. (n.d.). *EmoPy's documentation! — EmoPy 1.0 documentation*. Https://Emopy.Readthedocs.Io/En/Latest/. Retrieved April 5, 2021, from https://emopy.readthedocs.io/en/latest/

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2009). *The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Freund, Y., & Schapire, R. E. (1999). *Large margin classification using the perceptron algorithm. Machine Learning*, *37*(3), 277–296. https://doi.org/10.1023/a:1007662407062

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

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* (Vol. 1). Massachusetts, USA:: MIT press.

Google. (n.d*.). Colaboratory – Google. Google Colab. Retrieved* November 21, 2020, from https://research.google.com/colaboratory/faq.html

Gonzales, A. L., & Hancock, J. T. (2011). *Mirror, Mirror on my Facebook Wall: Effects of Exposure to Facebook on Self-Esteem. Cyberpsychology, Behavior, and Social Networking*, *14*(1–2), 79–83. https://doi.org/10.1089/cyber.2009.0411

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

Grasmuck, S., Martin, J., & Zhao, S. (2009*). Ethno-Racial Identity Displays on Facebook. Journal of Computer-Mediated Communication*, 15(1), 158–188. https://doi.org/10.1111/j.1083-6101.2009.01498.x

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

Guo, Y., Zhang, H., Gao, J., Wei, S., Song, C., Sun, P., & Qiao, M. (2015). Study of genes associated with the 'anger-in'and 'anger-out'emotions of humans using a rat model. *Experimental and therapeutic medicine*, *9*(4), 1448-1454.

Hamermesh, D., & Biddle, J. (1994). *Beauty and the Labor Market. The American Economic Review*, 84(5), 1174-1194. Retrieved September 18, 2020. from http://www.jstor.org/stable/2117767

Kaggle (2015) *Higgs Boson Machine Learning Challenge | Kaggle*. https://www.kaggle.com/c/higgs-boson

Image-net (2012). *ImageNet dataset*. Http://Image-Net.Org. http://image-net.org/challenges/LSVRC/2017/browse-synsets

Jackson, S. E., & Schuler, R. S. (1995). *Understanding human resource management in the context of organizations and their environments. Annual review of psychology*, *46*(1), 237-264.

Ji, Z., Sun, Y., Yu, Y., Guo, J., & Pang, Y. (2018). *Semantic softmax loss for zero-shot learning*. *Neurocomputing*, *316*, 369-375.

Jones, M., & Viola, P. (2003). *Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96*, *3*(14), 2.

kaggle. (2013, May 25). *Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle. Https://Kaggle.Com.* https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

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

Kobayashi, M. (2017). *Gradient descent learning for quaternionic Hopfield neural networks. Neurocomputing*, *260*, 174–179. https://doi.org/10.1016/j.neucom.2017.04.025

Kotikalapudi, R., Chellappan, S., Montgomery, F., Wunsch, D., & Lutzen, K. (2012). *Associating depressive symptoms in college students with internet usage using real Internet data. IEEE Technology and Society Magazine*, *31*(4), 73-80.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., & Ferrari, V. (2020). *The open images dataset v4. International Journal of Computer Vision*, 1-26.

Laisheng, X. (2014). *A Sliding-Window Modeling Approach for Neural Network. International Journal of Control and Automation*, *7*(8), 117–130. https://doi.org/10.14257/ijca.2014.7.8.11

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. https://doi.org/10.1080/00909882.2010.536844

Lienhart, R. (2002). *Classifying images on the web automatically. Journal of Electronic Imaging*, *11*(4), 445. https://doi.org/10.1117/1.1502259

Mahmood, H. (2018, November 27). *The Softmax Function Simplified - Towards Data Science. Medium*. https://towardsdatascience.com/softmax-function-simplified-714068bf8156

Markham, K. (2020, February 3). *Simple guide to confusion matrix terminology. Data School.* https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Marlowe, C. M., Schneider, S. L., & Nelson, C. E. (1996). *Gender and attractiveness biases in hiring decisions: Are more experienced managers less biased? Journal of applied psychology*, *81*(1), 11.

Méndez, N., Bouza, L. A., Chang, L., & Méndez-Vázquez, H. (2018). *Efficient and Effective Face Frontalization for Face Recognition in the Wild. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 391–398. https://doi.org/10.1007/978-3-319-75193-1_47

Meyer, D., & Wien, F. T. (2015). Support vector machines. *The Interface to libsvm in package* e1071, 28.

Mill, A., Kööts-Ausmees, L., Allik, J., & Realo, A. (2018). *The Role of Co-occurring Emotions and Personality Traits in Anger Expression. Frontiers in Psychology,* 9, 1–20. https://doi.org/10.3389/fpsyg.2018.00123

Mitchell, R., & Frank, E. (2017). *Accelerating the XGBoost algorithm using GPU computing. PeerJ Computer Science*, 3, e127. https://doi.org/10.7717/peerj-cs.127

Moreno, M. A., Jelenchick, L. A., Egan, K. G., Cox, E., Young, H., Gannon, K. E., & Becker, T. (2011). *Feeling bad on Facebook: depression*

disclosures by college students on a social networking site. Depression and Anxiety, 28(6), 447–455. https://doi.org/10.1002/da.20805

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.

Murase, H., & Nayar, S. K. (1995). *Visual learning and recognition of 3-d objects from appearance. International Journal of Computer Vision*, *14*(1), 5–24. https://doi.org/10.1007/bf01421486

NIST/SEMATECH. (2012, April). *Critical Values of the Student's t Distribution*. https://www.itl.nist.gov/div898/handbook/eda/section3/eda3672.htm

Pal, M. (2005). *Random forest classifier for remote sensing classification. International Journal of Remote Sensing,* 26(1), 217–222. https://doi.org/10.1080/01431160412331269698

Papageorgiou, C., & Poggio, T. (2000). *A Trainable System for Object Detection. International Journal of Computer Vision*, 38(1), 15–33. https://doi.org/10.1023/a:1008162616689

Pascal2 (2012). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012)*. *Oxford University*. http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

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

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 Arts.* https://www.thoughtworks.com/insights/blog/emopy-machine-learning-toolkit-emotional-expression

Pypi. (2021, January 30). *PyPI.* https://pypi.org/project/pip/

Pupale, R. (2019, February 11). *Support Vector Machines (SVM) — An Overview - Towards Data Science. Medium*. https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

Python Software Foundation. (2016, December 23). *Python 3.6*. Python 3.6. https://www.python.org/downloads/release/python-360/

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, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 1. https://doi.org/10.1109/cvpr.2016.91

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). *ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sánchez, J., & Perronnin, F. (2011). *High-dimensional signature compression for large-scale image classification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* June 2011, 1665–1672. https://doi.org/10.1109/CVPR.2011.5995504

Sadilek, A., Kautz, H., & Silenzio, V. (2012). *Modeling Spread of Disease from Social Interactions*. *Proceedings of the International AAAI Conference on Web and Social Media*, *6*(1). Retrieved from https://ojs.aaai.org/index.php/ICWSM/article/view/14235

Sasaki, Y. (2007). *The truth of the F-measure. School of Computer Science, University of Manchester*, 1–5. https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf

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

Selenium HQ. (2019, September 4). *ChromeDriver*. GitHub. https://github.com/SeleniumHQ/selenium/wiki/ChromeDriver

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.

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

Sheldon, K. M., Abad, N., & Hinsch, C. (2011). *A two-process view of Facebook use and relatedness need-satisfaction: Disconnection drives use, and connection rewards it. Journal of Personality and Social Psychology*, *100*(4), 766–775. https://doi.org/10.1037/a0022407

Steinfield, C., Ellison, N. B., & Lampe, C. (2008). *Social capital, self-esteem, and use of online social network sites: A longitudinal analysis. Journal of Applied Developmental Psychology*, 29(6), 434–445. https://doi.org/10.1016/j.appdev.2008.07.002

Stephen, I. (1990). *Perceptron-based learning algorithms. IEEE Transactions on neural networks,* 50(2), 179.

Syed, M. A. (2020). *Overview on Open Source Machine Learning Platforms-TensorFlow. SSRN Electronic Journal,* 1. https://doi.org/10.2139/ssrn.3732837

ThoughtWorksArts. (2018, September 6). *thoughtworksarts/EmoPy*. GitHub. https://github.com/thoughtworksarts/EmoPy

Utz, S., Tanis, M., & Vermeulen, I. (2012). *It Is All About Being Popular: The Effects of Need for Popularity on Social Network Site Use.*

Cyberpsychology, Behavior, and Social Networking, 15(1), 37–42. https://doi.org/10.1089/cyber.2010.0651

Waskom, M. (n.d.). *seaborn: statistical data visualization — seaborn 0.11.0 documentation. Seaborn Statistical Data Visualization.* Retrieved November 22, 2020, from https://seaborn.pydata.org

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.

Yu, A. Y., Tian, S. W., Vogel, D., & Chi-Wai Kwok, R. (2010). *Can learning be virtually boosted? An investigation of online social networking impacts. Computers & Education*, *55*(4), 1494–1503. https://doi.org/10.1016/j.compedu.2010.06.015

Zwanenburg, G., Hoefsloot, H. C. J., Westerhuis, J. A., Jansen, J. J., & Smilde, A. K. (2011). *ANOVA-principal component analysis and ANOVA-simultaneous component analysis: a comparison. Journal of Chemometrics*, 25(10), 561–567. https://doi.org/10.1002/cem.1400

8.Appendix

APPENDIX A

Representation of Resnet50 network, for classification of image features.

NetChain

	Input	image 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

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

APPENDIX B

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_di
mensions)+[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
```

Source code for emotions scoring, partially modified version of the Emopy package combining 2 setting to used in order to classify 4 emotions for each of the 2 standard models (https://github.com/thoughtworksarts/EmoPy), Perez, A. (2017)

APPENDIX C

Explanation of confusion matrix (Markham, 2020)

Simple guide to confusion matrix terminology

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

I wanted to create a **"quick reference guide" for confusion matrix terminology** because I couldn't find an existing resource that suited my requirements: compact in presentation, using numbers instead of arbitrary variables, and explained both in terms of formulas and sentences.

Let's start with an **example confusion matrix for a binary classifier** (though it can easily be extended to the case of more than two classes):

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

What can we learn from this matrix?

- There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 patients in the sample have the disease, and 60 patients do not.

Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

I've added these terms to the confusion matrix, and also added the row and column totals:

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

This is a list of rates that are often computed from a confusion matrix for a binary classifier:

- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?

  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"

- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
  - actual yes/total = 105/165 = 0.64

A couple other terms are also worth mentioning:

- **Null Error Rate:** This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be 60/165=0.36 because if you always predicted yes, you would only be wrong for the 60 "no" cases.) This can be a useful baseline metric to compare your classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate, as demonstrated by the **Accuracy Paradox**.
- **Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. (**More details about Cohen's Kappa.**)
- **F Score:** This is a weighted average of the true positive rate (recall) and precision. (**More details about the F Score.**)
- **ROC Curve:** This is 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 you vary the threshold for assigning observations to a given class. (**More details about ROC Curves.**)

APPENDIX D

Code for extraction of data from linkedin.

https://colab.fan/1dt9fybDrHsjEvvUx8VrD_tswRZJz-Wwz

APPENDIX E

Video for process of extracting picture from LinkedIn

thanakij wanavit. (2020, March 5). *extractPicsFromLinkedin* [Video]. YouTube. https://www.youtube.com/watch?v=N3sfYo867Sk