User Profiling in Social Media

TCSS 555: Machine Learning

Kenneth Goh
School of Engineering and Technology
University of Washington Tacoma
Tacoma Washington US
ggoh@uw.edu

Adilbek Dostiyarov
School of Engineering and Technology
University of Washington Tacoma
Tacoma Washington US
dostiyar@uw.edu

Sonia Xu
School of Engineering and Technology
University of Washington Tacoma
Tacoma Washington US
sxu253@uw.edu

ABSTRACT

Predicting user information based on individual users Facebook data has been widely studied in recent years. Novel methods relating to Natural Language Processing and Image Processing have been applied to predict gender and age using Facebook data.

We were provided user profiling data which we then categorized into three sections: status updates (text), profile pictures (images) and page "likes." We proceeded to apply machine learning methods ranging from Naïve Bayes classifier algorithm, K-Nearest Neighbors (kNN) classifier algorithm, Linear Regression, Logistic Regression, Ridge and Lasso regression with regularization, Random Forest classifier, Support Vector Machines (SVM), Convolutional Neural Networks (CNNs) and Neural Networks.

For page "likes," we found that logistic regression works the best in predicting gender and age.

For status updates (text) in predicting gender, we found that Naïve Bayes classifier algorithm and Random Forest classifier algorithm with n_estimator =100 has similar predictive accuracy of about 70%. Also, for predicting age, the Naïve Bayes classifier algorithm and SVM with kernel = 'sigmoid' has similar predictive accuracy of about 59%.

For profile pictures (image data), we found that CNN with 7 layers performs the best in predicting gender.

From the highest scoring training models for each category, we then built an ensemble method consisting of logistic regression on page "likes" data, random forest classifier and SVM on status updates (text) data, CNN on profile pictures (images) data, and neural network on the Big Five personality scores. The ensemble method used majority vote for deciding between the three training models for each input source for predicting gender.

Our results obtained from the ensemble method illustrate how accuracy can increase from diverse training model methods and demonstrates that estimators with high predictive capabilities on age, gender, and the Big Five personality scores is achievable.

When testing in the VM, we obtained 60% accuracy on predicting age, and 82% on gender. For the Big Five personality scores, RMSE of 0.65 for Openness, 0.81 for Neuroticism, 0.79 for Extraversion, 0.67 for Agreeableness and 0.72 for Conscientiousness.

KEYWORDS

Root Mean Square Error (RMSE), Accuracy, Natural Language Processing, Image Processing, Naïve Bayes classifier algorithm, K-Nearest Neighbors (kNN) classifier algorithm, Linear Regression, Logistic Regression, Ridge and Lasso regression with regularization, Random Forest classifier, Support Vector Machines (SVM), Convolutional Neural Networks (CNNs), Neural Networks, Ensemble method, Natural Language Toolkit (NLTK).

1 Introduction

In this paper, we are trying to predict gender and age of users which are multiclass classification problems. We also attempt to predict the Big Five personality scores of users which are regression problems.

Throughout this paper, we are evaluating our predictive models with an accuracy measurement, while the predicted Big Five personality scores are measured by Root Mean Square Error (RMSE).

Our approach takes input from Facebook users of the following:

1. Status updates (Text data)
2. Profile pictures (Image data)
3. Page "likes" (Things that users like)

Gender can be predicted as either "male" or "female", while age can be predicted as either "xx-24," "25-34," "35-49" or "50-xx." Lastly, the Big Five personality scores are predicted in a range between [1, 5].

The Big Five personality scores are studied extensively in several studies by Fabio Celli et al. [1]. Michal Kosinski et al. [2] has demonstrated in his study the predictive power of page "likes". Zhang et al. [3] has explained in his study of incorporating situation to predict personality in social media that the Big Five personality scores are known as Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism.

According to him, openness represents curious, intelligent and imaginative nature. Conscientiousness represents responsible, organized and persevering nature. Extraversion represents outgoing, amicable and assertive nature. Agreeableness represents cooperative, helpful and nurturing nature. Lastly, neuroticism represents anxious, insecure and sensitive nature.

Each week our training models were uploaded to a Virtual Machine where our training models were used to predict age, gender, and the Big Five personality traits based on test data. The results were posted to a scoreboard with a row filled with baseline values that our training models were expected to surpass eventually.

The baseline for the Big Five personality scores was calculated by taking the mean score based on the training data. The baseline for gender was based on taking the majority gender. The baseline for age was calculated by first preprocessing the age, into the desired age intervals and then taking the majority of the four age groups.

	Age	*Gender*	*O*	*C*	*E*	*A*	*N*
Baseline	0.59	0.59	0.65	0.80	0.79	0.67	0.73

Table 1: **This table represents the baseline scores calculated from the training data.**

1.2 The Dataset

Data was provided in two different sets of folders, a folder labeled "training" with 9500 users, and another folder labeled "public-test-data" containing 334 users. Both sets of folders were structured identically and both contained five subfolders labeled: "profile", "text", "image", "relation", and "LIWC."

The "profile" folder contains one csv file with 8 columns: "user id," "age," "gender," "ope," "con," "ext," "agre," and "neu." The data is represented as one row per user. The dataset in the "training" folder contains all the available values for each column. However, the dataset in the "public-test-data" folder only contains the listed user id values with all other column values empty.

The "text" folder contains one csv file per user which is labeled <userid>.txt and contains the combined status updates for each user. The status updates are mostly in English, but we were given notice that there may be some other languages mixed in.

The "image" folder contains one photo for each user id, labeled <userid>.jpg.

The "relation" folder contains a csv file with rows with two columns labeled: "userid" and "likeid." Each row indicates that the user (userid) liked the page (likeid). If one user liked 10 pages, there would be 10 total rows to convey this with each row displaying the userid and the associated like id.

The "LIWC" folder contains a csv file with one row per user. The columns of this csv file correspond to 82 LIWC features extracted from the status updates.

2 Methods

For each of the three input sources: of data: text, images and likes, we are tasked with creating a machine learning model that is able to predict variables for each type of data, with a higher accuracy than the given baseline. To accomplish this, we built individual machine learning models for each of the three different types of input sources and switched off testing the three different training models on the VM. Towards the end of the project, once we had established that each of the training models produced results that beat the baseline, we then proceeded to build ensemble models which combined different training models built from different input sources in order to experiment and see if we could achieve higher predictive accuracy and lower RMSE for the prediction of age and gender.

In total during the duration of this project we created four types of training models: training model for page likes as input, training model for text as input, training model for images as input, and heterogenous ensemble model.

2.1 Training Model for Page Likes as Input

To create a training model for page likes as input, we used the data from both the profile.csv file and relation.csv file located in the training folder. The profile.csv file contains user id, age, gender, openness, conscientiousness, extroversion, agreeableness, and emotional stability. The relation.csv file contains data for user id, and page likes which has 1,048,576 rows.

The first task in creating a training model for page likes is to preprocess the given data to be formatted in a useable structure. This is accomplished by organizing the information into a dictionary data structure, with the key as the user id and the value as a compilation of all the page like id's associated with the specified user id. This combines the necessary information to then create an entirely new csv file and merge all the existing columns in the profile.csv with the page like values. The new csv file will have a column with all the newly compiled page likes for each unique user id. The page like ids are separated by a space and inserted into one cell for each user id. So for each row there now exists a column that contains all the pages a certain user has liked.

For training the model, the cell of page likes separated by a space was treated as a group of text. Next, the data was transformed into an acceptable format in order for machine learning algorithms to work. Therefore, we implemented Bag-of-Words Model, using count vectorizer. This model discards the order of how the information is presented and focuses instead on the occurrences of words, in our specific case the page like id's in the csv file for the "likeid" column. This is accomplished be giving each word a unique number and creating vectors with the length of the number of unique page ids.

2.1.1 Methodology Naïve Bayes, KNN, and Logistic Regression were applied to the likes model for age and gender. Linear regression was applied to the big five personality traits.

To implement Naïve Bayes, KNN, and Logistic Regression, count vectorizer is used to build a list of all unique page likes using the fit() function, and then using the transform() function to encode each user id's column of compiled like id's as a vector. The transform() function returns an encoded vector that is the length of the all the unique page likes from the given data and an integer count for the number of times each page like appeared in the file. The return vector is what we then used as input for each machine learning model to train the data.

The evaluation measure for age and gender is the accuracy between the target output and the predicated output. The root mean squared error measurement was used to evaluate the five personality traits. Training models were uploaded to the VM using Pickle.

2.2 Training Model for Status Updates as Input

To create a training model for Status Updates as input, we used the data from the text folder which contains one csv file per user for the combined status updates for each user. and profile.csv file which contains user id, age, gender, openness, conscientiousness, extroversion, agreeableness, and emotional stability.

There was a total of 9500 csv files for each user from the text folder and 9500 rows in the profile.csv file.

2.2.1 Methodology Naïve Bayes classifier, Random Forest classifier, Support Vector Machine with the sigmoid as kernel were applied to the text model for age and gender. Linear regression, ridge regression with regularization, lasso regression with regularization, random forest regression, neural network with ADAM optimizer was applied to the big five personality traits.

Individually, Naïve Bayes classifier algorithm was applied to predict the gender and age of each user. First, we did this by first extracting 'text' column from each of the users and then combining the text rows to its respective users. In doing this, I make sure to convert the ages of each respective users into their respective age groups, "xx-24", "25-34", "35-49" or "50-xx". Then I used the training data consisting of 'userid', 'age', 'gender' and 'text' columns to train a Naive Bayes model to predict gender and age.
After which, we also used linear regression for the Big five personality scores, agr, neu, ope, neu and ext values.

Finally, we managed to populate xml files for each user with predicted 'gender', 'age', agr, neu, ope and ext values.

On our Ubuntu VM, we managed to obtain 0.59 and 0.71 for the accuracy for the prediction of age group and gender respectively. For the Big Five personality scores, we obtained RMSE of 0.65, 0.79, 0.79, 0.68 and 0.72 for ope, neu, ext, agr and con respectively.

	Age	Gender	O	N	E	A	C
Week 4	0.59	0.71	0.65	0.79	0.79	0.68	0.72

Table 2: **This table represents the scores of the text model in week 4. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

Then, we also managed to use Pickle to save trained models and tested by using Pickle to load trained models as well. These are all aims from the in-class Project Update 1. It was at this point that we decided to try the LWIC personality scores with Regularization using ridge regression and lasso regression was tried on our own Anaconda Spyder environment with the following specifications:

Processor	RAM	System type	OS
i7-8750H CPU, 2.20GHz	8.00 GB	64-bit OS, x64-based	Windows 10 Home

Table 3: **This table represents the specifications of the system on which the ridge and lasso with regularization was ran on.**

Then we obtained the results on the above environment:

	Ope	Con	Ext	Agr	Neu
Baseline	0.65	0.73	0.79	0.66	0.80
Lasso regression	0.62	0.73	0.81	0.65	0.79
Ridge regression	0.62	0.73	0.83	0.69	0.78
VM Scoreboard	-	-	-	-	-

Table 4: **This table represents the specifications of the system on which the ridge and lasso with regularization was ran on. More specifically, we ran GridSearchCV from sklearn.model_selection with alpha parameters from 1 to 20 with increment of one and cv=5.**

The results in Table 4 looks promising for predicting ope, con and agr values using Lasso regression, and also using ridge regression to predict ope and con values. Moreover, the results were also presented in an update.

Then, we proceed to use Natural Language Toolkit (NLTK) for gender and age prediction and pre-processed the text data by **removing the top 20 most commonly used words and the least 88000 used words**. **The time taken for pre-processing this time around took approximately 8 hours on the system with specifications mentioned in Table 3.**

After which, we applied ensemble methods such as **Random Forest regression on the Big Five personality scores.** We also tried applying deep learning techniques which was one of our aims from one of our progress updates. Specifically, we used **Random Forest Classification with n_estimators = 400, bootstrap = true and max_features = 'sqrt' to predict gender**. For age prediction, it was **Support Vector Machine SVM** with **kernel = 'sigmoid'** and **gamma =0.007**. For the **Big Five personality scores**, we used **model.add(Dense(3,input_dim=82,activation='sigmoid')) once**, **model.add(Dense(3,input_dim=82,activation='selu')) twice** and **model.add(Dense(3, input_dim=82, activation='relu')) twice,** and ending with the **ADAM optimizer**. We used **epochs = 60 for agr, ope and con, epochs = 50 for neu**, and finally **epochs = 90 for ext.** But this resulted in an overfitting case as we did not properly controlled for the testing this time round.

The results of running in our Ubuntu VM environment are as follow:

Attempt	Age	Gender	O	N	E	A	C
1	0.50	0.50	0.65	0.81	0.79	0.65	0.72

Table 5: **This table represents the 1st attempt of the text model in week 10. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

Then, we attempted another approach in which **we ran the Random Forest Classifier algorithm for the prediction of gender with n_estimators = 100**. Then we implemented **Support Vector Machine SVM algorithm for the prediction of age** with **kernel = 'sigmoid'**. However, the previously trained neural network models for predicting agr and ope values are returning very big values when we ran it on the same validation test set. So, we **retrained the neural network models for predicting agr and ope values using epochs =50 for agr and epochs = 60 for ope**. We reused the neural network previously trained to predict the neu, ext and con values.

Upon testing on the test data on our Ubuntu VM again, we achieved the following results:

Attempt	Age	Gender	O	N	E	A	C
2	0.59	0.69	0.65	0.81	0.79	0.67	0.72

Table 6: **This table represents the 2nd attempt of the text model in week 10. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

We think the **significant improvement in the prediction accuracy for the age and gender** is most probably due to the fact that we did not do pre-processing on the text by removing the top 20 common and least 88000 words like the first attempt. It is probably that the number of words to be removed, whether top n common words or least m common words to improve prediction accuracy actually involved more testing in order to reduce overfitting.

Another method to be tried on the LIWC values was **Random Forest Regression**. We trained and deployed it with **n_estimators = 300 for predicting agr, and neu values** and **for predicting con and ext values**, we used **n_estimators = 100**. Lastly for **predicting ope values, we used n_estimators = 200. Then running it on the system specified in Table 3, we obtained the following results:**

	O	C	E	A	N
Baseline	0.65	0.73	0.79	0.66	0.80
Random Forest	0.61	0.72	0.81	0.66	0.80

Table 7: **This table represents in predicting the Big Five Personality Scores in week 10. These scores were deployed on the system as specified in Table 3.**

2.3 Training Model for Status Images as Input

Convolutional Neural Networks is well known technique for classification based on images. This project also required use of CNN for predicting gender and age of a user. Data preprocessing for images is the same for age and gender.

First, we transform RGB image into a grayscale in order to avoid creation of inappropriate features based on the color of the image.

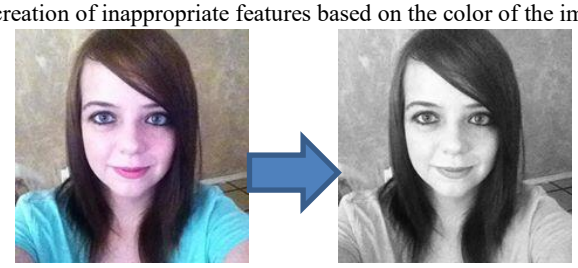

Figure 1. turning RGB image into a grayscale.

Then we crop face from an image because other objects on an image could also create unnecessary features during the training process. We tried different techniques for cropping face out of image. These are Haarcascades and face_crop method from cvlib library.

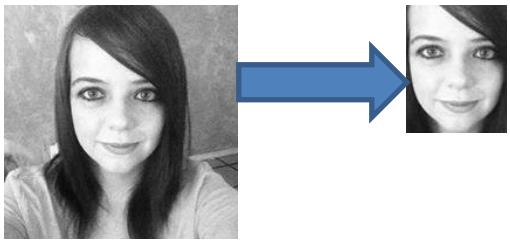

Figure 2. Cropping face from initial image

We decided to use face_crop because it provided better cropping abilities on this type of data.

Our next step was to resize face images so that the input data have the same parameters. First we worked with 96 by 96 images, but then we decided to work with 64 by 64 due to the fact the accuracy does not suffer but training speed increases.

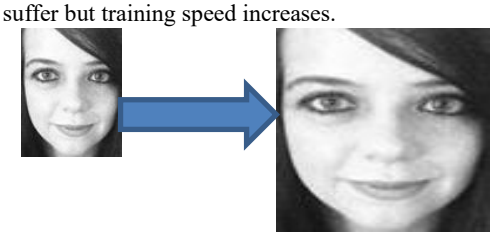

Figure 4. Rescaling face to a unified size

Moreover, in order to balance data we also created additional images based on a cropped face. This technique allowed us to create artificial data that we also fed into our training model. We rotated image based on how much we needed to balance data. For example, for the third age group we rotated our image by 5, 10 and 15 degrees to make dataset for age-group "35-49" four times bigger.

Figure 5. Rescaling face to a unified size

Final step for preprocessing was to convert image into array and add it to an input list.

```
[[204.][201.][199.]
 ...
[139.][134.][129.]]
 ...
[[147.][149.][151.]
 ...
[190.][188.][184.]]
```

Figure 6. transforming image into array

Labels were easily extracted from profile.csv file using the name of an image.

Training CNN model may vary on several parameters: number and type of layers, activation functions, kernels and etc. We tried a number of different CNN architecture and the one proposed below performs better compared to others:

```
Conv2D(32, (3,3))
Activation("relu")
BatchNormalization
MaxPooling2D(3,3)
Dropout(0.25)
Conv2D(64, (3,3))
Activation("relu")
BatchNormalization
Conv2D(64, (3,3))
Activation("relu")
BatchNormalization
MaxPooling2D(2,2)
Dropout(0.25)
Conv2D(128, (3,3))
Activation("relu")
BatchNormalization
Conv2D(128, (3,3))
Activation("relu")
BatchNormalization
MaxPooling2D(2,2)
Dropout(0.25)
Conv2D(256, (3,3))
Activation("relu")
BatchNormalization
Conv2D(256, (3,3))
Activation("relu")
BatchNormalization
MaxPooling2D(2,2)
Dropout(0.25)
Dense(1024)
Activation("relu")
Dense(#Classes)
Activation("softmax")
```

We used the same architecture for age and gender that differs with the last dense layer that contains number of classes (2 for gender, 4 for age) as an input parameter.

Moreover, we varied different parameters during the fitting the model. Finally we decided to use 100 epochs, batch size = 64, and loss = binary_crossentropy for gender and categorical_crossentropy for age.

2.4 Building the Ensemble Model as Input

We decided to build an ensemble model from our three different independent training models for predicting age group and gender. We implemented trivial majority out of three for gender and age. After testing our first ensemble on the VM, we later identified that age did not perform well and decided to create another version of the ensemble model for gender only.

3 Results

3.1 Results for Likes Model

	Age	*Gender*	*O*	*C*	*E*	*A*	*N*
Baseline	0.59	0.59	0.65	0.80	0.79	0.67	0.73
Week 5	0.59	0.71	0.65	0.80	0.79	0.67	0.73
Week 6	0.61	0.75	0.65	0.80	0.79	0.67	0.73
Week 9	0.66	0.81	0.65	0.80	0.79	0.67	0.73

Table 8: **This table represents the scores of the page likes model and shows specific weeks where a milestone was achieved with an increase in accuracy. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

The first machine learning model used to train the likes model and deployed on the VM was Naïve Bayes. We first preprocessed the data to filter out pages with less than ~2-25 likes and more than ~900-1000 likes in attempts to create a meaningful data set. The filtration of page likes was based on the idea that people who like nothing would not contribute meaningful information and people who liked everything would not be able to contribute meaningful information. The initial model was trained using Naïve Bayes with this page like filtration and the results on the VM is shown as week 5 in table 1. For this particular model, the predicated age accuracy remained the same as the baseline, while the gender accuracy increased to 0.71 from 0.59.

In attempts to increase age accuracy, we decided to use KNN to train the model. After adjusting the number of neighbors a few times, the highest accuracy that could be achieved at the time for age was still 0.59 at ~250nn. The accuracy for gender was similar to the accuracy I received for Naïve Bayes. KNN did not produce significant results and it was decided we would not move forward with finetuning the model. KNN did not produce significant results and this could be a few reasons. The data set could not be easily separable. The data set might not be sufficient enough in training examples because the KNN algorithm relies on the assumption that similar things exist in close proximity.

To continue on the pursuit of increasing accuracy, we decided to keep using Naïve Bayes but this time, we eliminated the before mentioned page like filtering. After discussing with other team members who were responsible for the likes model, it was discovered that those who had not been implementing any page filtering received higher results than what was scored initially.

After deploying the model using Naïve Bayes with the full original data set, the accuracy for age increased from 0.59 to 0.61 and gender increased from 0.71 to 0.75. Our conjecture for the accuracy increase is the more information, the better learning the training the model can do. Because we are given less than 10,000 users, even page likes that aren't liked by many or that are liked by all could still provide some meaningful information because of our lack of ample information.

It's worth mentioning that at this time when we eliminated page filtering, we discovered after speaking with a teammate that the preprocessed data had been created with a corrupted Excel file, meaning the dataset contained only information for ~6700 users and not the full 9500 users the training data set is expected to contain. With this in mind, we redownloaded the original Excel file and confirmed the program reads through all 9500 users. This further increases our conjecture that the less training data for an already small sample size, the probability it is likely to decrease the amount of potential learning for a training model.

At this time, we also decided to implement linear regression on the 5 personality traits. There was no significant change in results. At this point we decided to focus our time on continually improving gender and age.

The last machine learning algorithm implemented that gave the highest accuracy result since the start of the project was logistic regression. Logistic regression was the machine learning method used for week 9 on the VM and the results increased from 0.75 to 0.81. Logistic regression gave the best results out of all the machine learning algorithms implemented for several reasons. Logistic regression performed the best on this set of data for a few reasons. First, logistic regression does not assume a linear relationship between independent variables and dependent variables, to simplify it does not assume equal weight between input results and the output result. Logistic regression is also robust to noisy data. Logistic regression could have also performed well on our model because this algorithm determines well-calculated predicted probabilities which it then uses to provide the final classification for an input.

To summarize for the likes training model, we were able to get the results from the likes model increased from the baseline of 0.59 for age to 0.66 and 0.59 for gender to 0.81.

3.1.1 Future Considerations For future considerations, we would put more effort into lowering the root mean squared error for the big five personality trait. This would be done by implementing lasso regression and random forest regression. With more time we would also like to try to create an ensemble method within the likes model for gender since we created three different training models each using a different machine learning method that performed above the baseline.

3.2 **Results for Text Model**

The results from running the various text models are shown in the table below:

	Age	*Gender*	*O*	*C*	*E*	*A*	*N*
Baseline	0.59	0.59	0.65	0.73	0.79	0.66	0.80
Week 4	0.59	0.71	0.65	0.72	0.79	0.68	0.79
Week 10	0.50	0.50	0.65	0.72	0.79	0.65	0.81
Week 10	0.59	0.69	0.65	0.72	0.79	0.67	0.81

Table 9: **This table is a summary of the 3 attempts of the text model throughout week 4 to 10. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

The methods used above and ran in the VM are summarized below:

	Age	*Gender*	*O*	*C*	*E*	*A*	*N*
Week 4	NB	NB	LR	LR	LR	LR	LR
Week 10	SVM-sigmoid - 0.007	RF-400	NN-60	NN-60	NN-90	NN-60	NN-50
Week 10	SVM-sigmoid	RF-100	NN-60	NN-60	NN-90	NN-50	NN-90

Table 10: **This table is a summary of the methods used in the 3 attempts of the text model throughout week 4 to 10. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

For the text models tested on the VM, we found that both the Naïve Bayes and Random Forest Classifier with n_estimator=100 performs the best when it comes to classifying gender. For age classification, both Naïve Bayes and SVM performs moderately.

Lastly for the Big Five Personality Scores, we found that Linear regression works for both con and neu values in terms of RMSE. For both con and agr values, neural network with epochs=60 works as it improves the two values in terms of RMSE as well.

The results from running other methods are shown in the table below:

	Ope	Con	Ext	Agr	Neu
Baseline	0.65	0.73	0.79	0.66	0.80
Lasso regression	0.62	0.73	0.81	0.65	0.79
Ridge regression	0.62	0.73	0.83	0.69	0.78
Random Forest Regression	0.61	0.72	0.81	0.66	0.80

Table 11: **This table is a summary of the methods used in the 3 attempts of the text model throughout week 4 to 10. These scores were deployed on the system specified in Table 3.**

The details of the implementation are as follow:

	Ope	Con	Ext	Agr	Neu
Lasso regression	GridSearchCV, alpha=1 to 20, cv = 5	GridSearchCV, alpha=1 to 20, cv = 5	GridSearchCV, alpha=1 to 20, cv = 5	GridSearchCV, alpha=1 to 20, cv = 5	GridSearchCV, alpha=1 to 20, cv = 5
Ridge regression	GridSearchCV, alpha=1 to 20 and 30, cv = 5	GridSearchCV, alpha=1 to 20 and 30, cv=5	GridSearchCV, alpha=1 to 20 and 30, cv=5	GridSearchCV, alpha=1 to 20 and 30, cv=5	GridSearchCV, alpha=1 to 20 and 30, cv=5
Random Forest Regression	n_estimators = 200	n_estimators = 100	n_estimators = 100	n_estimators = 300	n_estimators = 300

Table 12: **This table summarizes the details of the implementations used in the 3 attempts of the text model throughout week 4 to 10. These scores were deployed on the system specified in Table 3.**

From the above two tables, we can see that lasso regression improves the ope, agr and neu scores. While we can observe that ridge regression improves the ope and neu scores. Lastly, the results also show us that random forest regression improves the ope, con and agr values.

In analyzing the results obtained from the various text models, we think that Naïve Bayes classifier algorithm works the best for predicting gender and age since it works based on probabilities and also there is not much hyper-parameters to tune when deploying it. However, the other methods which are Support Vector Machines and Random Forest Classifier algorithms will be more time-consuming to deploy since we have to further tune the number of trees or n_estimators in order to achieve better accuracy for predicting the gender and age of users.

A similar explanation can also be true for the Big Five Personality scores, Linear Regression works best as it is less time-consuming to deploy abut both ridge (L2 regularization) and lasso (L1 regularization) regression offers promising results if we are given more time to tune the penalty term, which is alpha. **GridSearchCV** implementation also offers to automatically find the 'best' alpha value in Python implementation.

In contrast, when trying to tune the random forest algorithm for regression and classifying, unless we chose to use implementations such as **Random Hyperparmeter Grid for Python** from the start, we would also need to spend a fair bit of time finding the 'best' hyperparameters. But both ridge and lasso regression will have lesser hyperparameters to tune.

3.3 Results for Images Model

	Age	*Gender*	*O*	*C*	*E*	*A*	*N*
Baseline	0.59	0.59	0.65	0.80	0.79	0.67	0.73
10/25	0.59	0.72	0.65	0.80	0.79	0.67	0.73
Week 8	0.59	0.73	0.65	0.80	0.79	0.67	0.73
12/01	0.60	0.82	0.65	0.81	0.79	0.67	0.73

Gender prediction based on images was the first evaluated model. We implemented 5 layer CNN based on VGG net architecture. In that implementation we used majority gender for the images that contain more than 1 person. However, when we put the same architecture on a VM with the random taking face out of image and evaluating gender based on this face the accuracy increased by 0.73.

We also tried to increase accuracy for age prediction using image as a source. We used many different architectures. However, the official results for age prediction were lower than a baseline. We also tried to use pre-trained caffe model.

Even though the classes for caffe model were different: ['(0, 2)', '(4, 6)', '(8, 12)', '(15, 20)', '(25, 32)', '(38, 43)', '(48, 53)', '(60, 100)'] we were still be able to predict age by merging the groups like (0, 2), (4, 6), (8, 12), (15, 20) into 'xx-24'. This approach did not help us to increase the accuracy for predicting age group based on image. Here is a graphic representing

Figure 7. Training accuracy, validation accuracy, training loss and validation loss while training model for 100 epochs.

3.3 Results for Ensemble Model

We created our ensemble training model from three different training models based on different input sources. The ensemble method uses majority vote for deciding between the three training models for each input source for predicting age and gender.

For the likes model, we implemented logistic regression for predicting age and gender.

For the text model, we implemented random forest classifier with n_estimators=100 for predicting gender and SVM with kernel='sigmoid' for predicting age and gender.

For the images model, we used final CNN architecture with 7 layers based on VGG Net with 2 classes for gender detection. And the same architecture with 4 classes for age group prediction.

We ran the heterogenous ensemble model consisting of the likes, text and images training models on the test data residing on the VM. From the evaluation which can be seen in Table 13 attempt 1, the results showed us a slight increase in our gender accuracy while it also showed us a significant decrease in our age accuracy. We believe our age accuracy decreased dramatically because two of our training models only did slightly better than the baseline, while the other model showed an adequate accuracy. Therefore, we decided to predict age using only our likes model which is the best performer for predicating age-group. The results for this ensemble method can be seen in Table 13, attempt 2.

For the Big Five Personality scores, we implemented the neural network models, which we also implemented in the 2nd attempt in week 10 (Table 10).

The results of the heterogenous ensemble model ran in our Ubuntu VM environment.

Attempt	Age	Gender	O	N	E	A	C
Baseline	0.59	0.59	0.65	0.80	0.79	0.66	0.73
1	0.60	0.82	0.65	0.81	0.79	0.67	0.72
2	0.66	0.81	0.65	0.81	0.79	0.67	0.72

Table 13: **This table summarizes the results obtained from running the heterogenous ensemble model on the VM. These scores were deployed on the virtual machine and therefore verified that the scores represent the accuracy of the training models deployment on new test data.**

4 Conclusion

Overall, each of our training models had better results when predicting gender. This is to be expected because the probability of getting it correct by just guessing is the best with a 50/50 chance. Predicting age was harder for text and images, while the likes model was able to beat the baseline adequately. Predicting personality scores provided to be a challenge for all sources.

In all, we learned that ensemble methods have the ability to outperform standalone models when the models used for the ensemble are able to produce results adequality above the baseline. We have learned through experience that it's true that machine learning models can do better through diversity.

We also discovered over these past few weeks just how challenging it can be to create machine learning models that have a high degree of accuracy. Through this we also learned the value of having large samples of data and the struggles of creating models that overfit the data.

ACKNOWLEDGMENTS

Professor Martine DeCock provided valuable guidance throughout this project and gave suggested options for which machine algorithms to try out. We will like to offer our heartfelt thanks to her.

REFERENCES

[1] Fabio Celli, Elia Bruni, Bruno Lepri. Automatic Personality and Interaction Style Recognition from Facebook Profile Pictures. Proceedings of the 22nd ACM International conference on Multimedia.

[2] Michal Kosinski, David Stillwell, Thore Graepal. Private traits and attributes are predictable from digital records of human behaviour. Proceedings of the National Academy of Science 110(15), March 2013.

[3] Lei Zhang, Liang Zhao, Xuchao Zhang, Wenmo Kong, Zitong Sheng, Chang-Tien Lu. Situation-Based Interpretable Learning for Personality Prediction in Social Media. 2018 IEEE International Conference on Big Data.

[4] Mitchell, Tom Michael. *Machine Learning*. McGraw-Hill, 1997.

[5] *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[6] Ito, Koichi, Hiroya Kawai, Takehisa Okano, and Takafumi Aoki. 2018. "Age And Gender Prediction From Face Images Using Convolutional Neural Network". APSIPA Annual Summit And Conference 978-988 (-14768-5-2): 7-11.

[7] Levi, Gil, and Tal Hassncer. 2015. "Age And Gender Classification Using Convolutional Neural Networks". 2015 IEEE Conference On Computer Vision And Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw.2015.7301352.