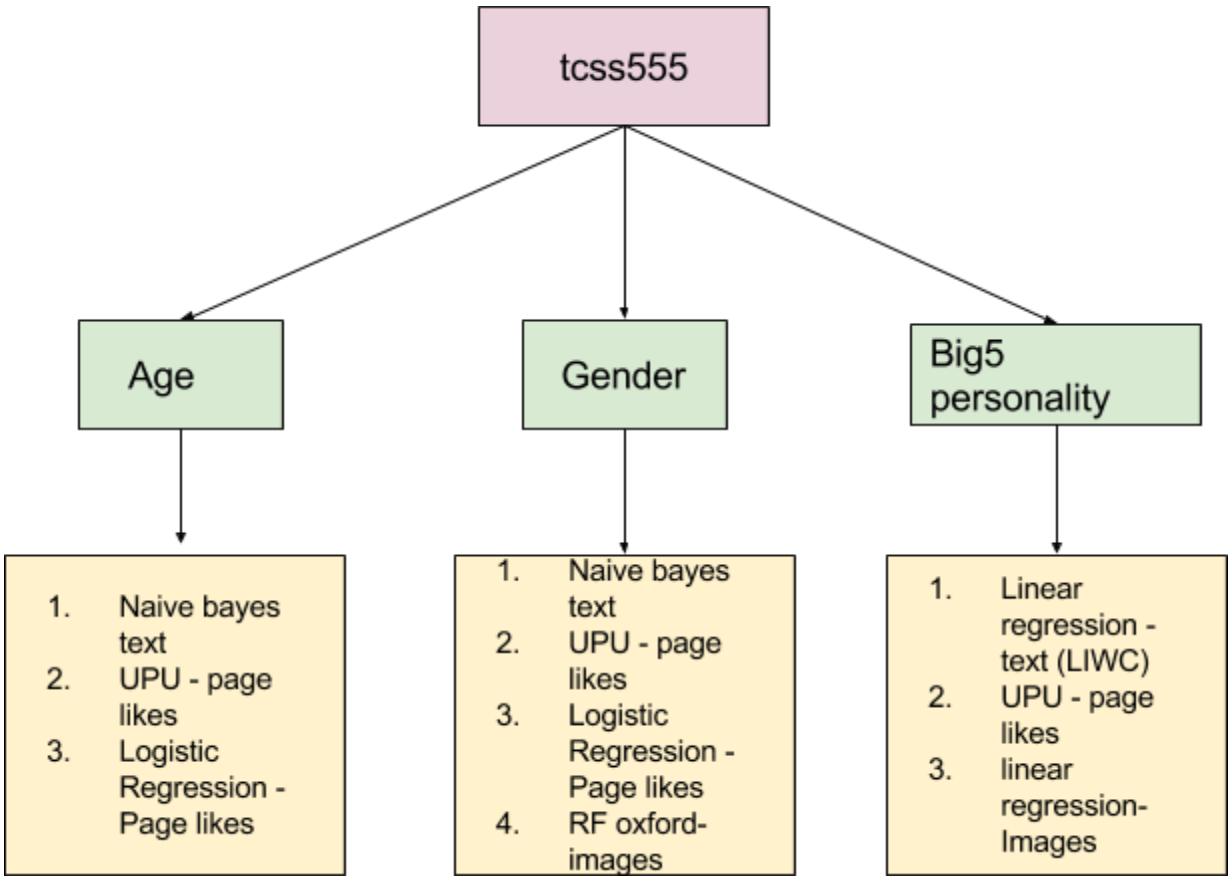
ReadMe

Kebra Thomson, Swati Garg and Surabhi Agrawal

This document provides implementation details about the scripts used to predict the age, gender and big5 personality traits using different machine learning models.



tcss555:

This is the runner for the complete project. This runner takes the input and output file paths as input. The command to run tcss555 is as follows:

tcss555 -i /data/public-test-data -o output/

1. The tcss555 first trains the models for Images, Text and Relation data individually. The models used for prediction are:
    - User page user -page likes
    - logistic regression - page likes
    - naive bayes - text
    - linear regression - text
    - random forest - images
    - linear regression - images

2. Once the models are trained, the unseen data is provided to these models to predict the age, gender and personality for each user.
3. The age bucket and gender are calculated from each model for each unseen user, and the majority age and gender value is predicted for the user.
4. The values of each of the big 5 personalities are predicted using these models individually, then their average value is the predicted value for the unseen user.
5. The xml file is written for each user with the file name *<userid>.xml.* The xml file is in the following format

    <User id = "f67e60570d27984cd729aa128e6d9b08"
    age_group = "25-34"
    gender = "Female"
    open = "3.89886155017"
    conscientious = "3.57752855848"
    extrovert = "3.61433069746"
    agreeable = "3.70635219047"
    neurotic = "2.68082332498"
    />
6. The libraries used in this runner class are:
    pandas
    numpy
    Collections

AgeGenPred.py:

1. This is the script to predict the age and gender using text data with naive bayes classifier. The script uses naive bayes classifier.

2. The text files for each user is read and converted into the bag of words using the nltk library
3. The stopwords are removed from the bag of words
4. The freq of each word is calculated and stored as tuple
5. The features are selected using the most common 500 features from the tuples.
6. The model is trained using the naive bayes classifier
7. The above process is repeated for the unseen user data to form the bag of words
8. The age and gender is predicted.
9. Libraries used:
    a. NLTK
    b. numpy
    c. scikit-learn.org

PerPred.py:

1. This script is used to predict the big5 personality traits for the text data using the linear regression
2. The script uses the LIWC features for the user given
3. The data frame is created using pandas to combine the LIWC and user profile data
4. The model is trained using linear regression from sklearn library
5. above process is repeated for the unseen user data to create the dataframe using the “LIWC” features.
6. libraries used:
    a. scikit-learn.org
    b. pandas
    c. numpy

page_likes.py:

1. This script uses ‘page likes’ information to predict the age, gender and personality scores for unseen users.

2. The technique used is the ‘User-Page-User’ approach. In this technique, we maintain a dictionary of the users who like each page. Eg: {Page1: [User1, User5], Page2: [User5, User6].....}. When we encounter a new instance of the form User10 : [Page 2, Page7, Page9], we obtain the class labels for each of the users who have liked the same pages as the unseen user, and take a majority of those values for predicting age and gender (classification problems) and average for personality traits (regression problems) for prediction.
3. The thresholds 0 and 50 in the script indicate that we have selected only those pages who have between 0 and 50 users liking that page. This is a form of feature selection for eliminating pages which do not give us much information about the class labels.

log_reg_build.py:

1. This script uses ‘page likes’ information to predict the age and gender for unseen users.
2. The technique used is Logistic Regression.
3. This script specifically just trains the Logistic Regression model and saves it to the ‘model’ folder.
4. The thresholds 4 and 200 in the script indicate that we have selected only those pages who have between 4 and 200 users liking that page. This is a form of feature selection for eliminating pages which do not give us much information about the class labels.
5. Specifically, the input given to the Logistic Regression is of the form:
   X = [[0 1 0 1 1 0 0 …….]
   [0 0 0 1 1 0 1……..]
   ……..]
   Yage = [0 1 3 2 …….]
   Ygen = [0 1 1 0 ……..]

where, X is the input matrix and each row represents a user and the 0 and 1 represent whether that user has liked that particular page. Yage is the label vector for age prediction where the values are the numbers given to the age buckets. 0→ xx-24 , 1→ 25-34, 2→ 35-49, 3→ 50-xx.
Ygen is the label vector for gender prediction where 0→ Male and 1→ Female.

5. This scripts trains two models, one for age and one for gender.

log_reg_query.py:

1. This script queries the trained model from the Logistic Regression (**log_reg_build.py**) to predict age and gender.
2. This script first reads the trained models for age and gender, using the 'joblib' library.
3. Then this script obtains the userid from the runner script and passes along the pages liked by that user by converting it to the form,
   [0 1 1 0 0 1 1 …….]
4. The above vector represents which pages the unseen user has liked or hasn't.
5. Then the script queries the trained models with this vector to obtain the age and gender predictions.

oxford.py:

1. This script trains the models used based on images. It returns a list of 7 models, age prediction, gender prediction, openness, as well as the other four personality models.
2. The oxford.csv file is read in during the script. This file contains the oxford features based on the face(s) discovered from their profile pictures. Not all users have a face that was detected. If not, they do not have any oxford features. Some users may have more than one face detected in their profile picture. In this case, the first photo detected is used.
3. The script reads in all of the oxford features and makes list of the features and their class label (an age bucket, gender, or personality score). A few features that were mostly 0's across the users are eliminated.
4. It then creates age and gender classifiers using a random forest model with 10 trees.
5. The script also creates the five separate personality prediction models using linear regression on the oxford features.
6. The 7 models are returned to the runner script.
7. The oxford.py file also contains a function which is used for prediction. This function is called from the runner, tcss555. It takes the user id as input, finds the oxford features of the user, and then predicts each of the 7 values for the user.

8. If the user id has no oxford features, the script returns the most common gender and age values (‘female’ and ‘xx-24’) as well as the average personality scores.
9. These values are then used in tcss555 along with the predictions from the other models to make the official prediction.