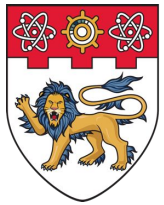


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Gender Prediction Based on Profile Photo

CE9010 Project -- Group 10:

Chen Zitong, Jin Ye, Xiao Fengtong

Outline

Part I. Problem Description
Part II. Data Acquisition
Part III. Data Exploration
Part IV. Data Preprocessing
Part V & VI. Data Analysis and Result Analysis
    i. SVM
    ii. Logistic Regression
    iii. Neural Network
    iv. Boosting
    v. Test on additional dataset
Part VII. Future Work

Github Repository:
https://github.com/FengtongX/CE9010_project

Python Notebook running on ***Google Colaboratory***

Part I. Problem Description

We aim to learn a binary-classification model to predict males and females based on their frontal face images.

Part II. Data Acquisition

Part 1:

We downloaded* 946 photos from MIVIA LAB with 255*383 pixels, then we resize the photo into 76*144 = 8664 pixels to save computational cost.

Further, we load the images and save it as .mat file on Google Drive to run on Colab notebook.

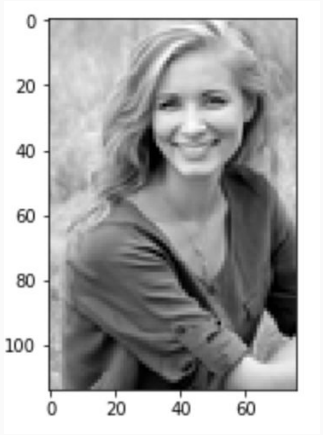
`dataset.shape = 8665 * 946` (label added)

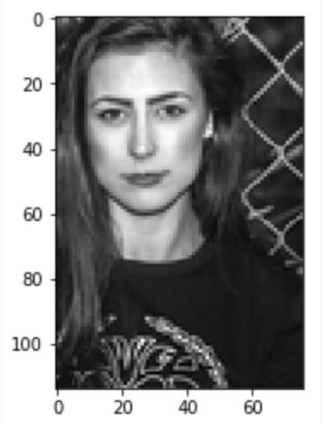
*LInk:http://mivia.unisa.it/datasets/video-analysis-datasets/gender-recognition-dataset

Part II. Data Acquisition

Part 2:
Scrape profile photo found online without copyright through our own scraper, further resized to 76*144 while keep face in the center.

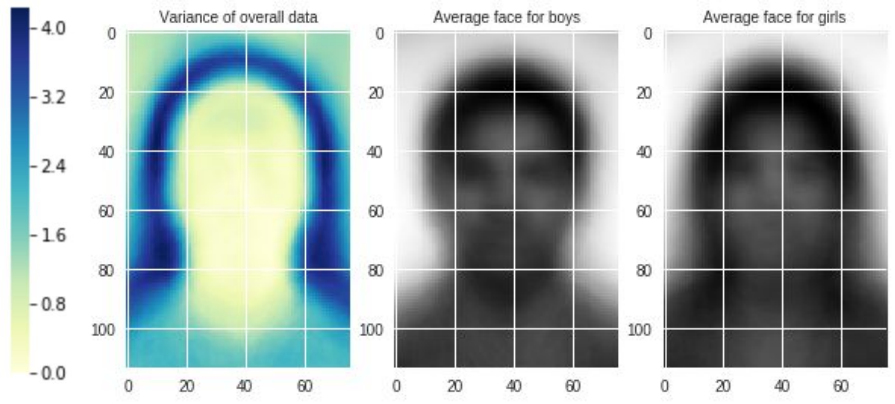
```
add_test_data.shape = 8665 * 37
```



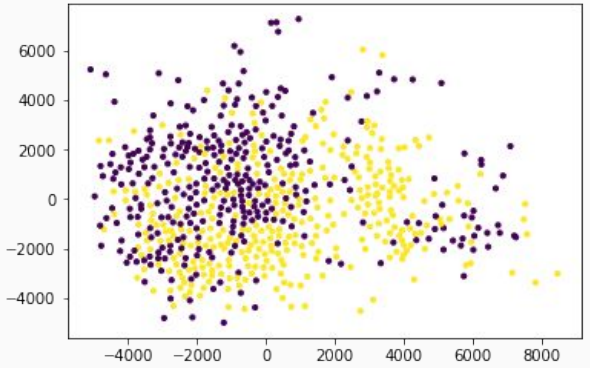


Part III. Data Exploration

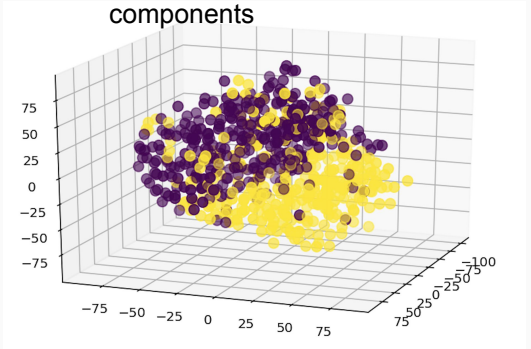
- **Mean & Variance**



- **Data compression using T-sne**



Reduced to 2 components



Reduced to 3 components

Part IV. Data Preprocessing

- **Splitting dataset**

```
train,test = train_test_split(dataset.T,test_size=0.3, random_state=42)
```

We split the dataset into 70% training set (622) and 30% testing set (384)

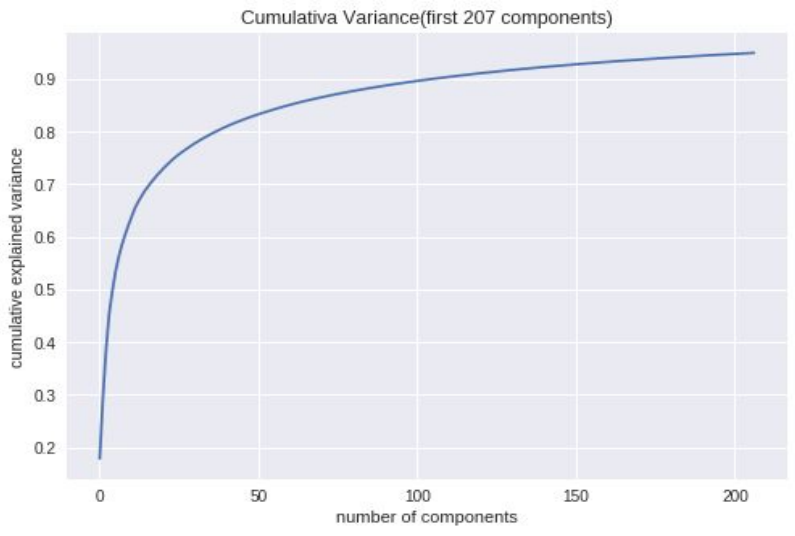
- **Normalization: *Z* - score**

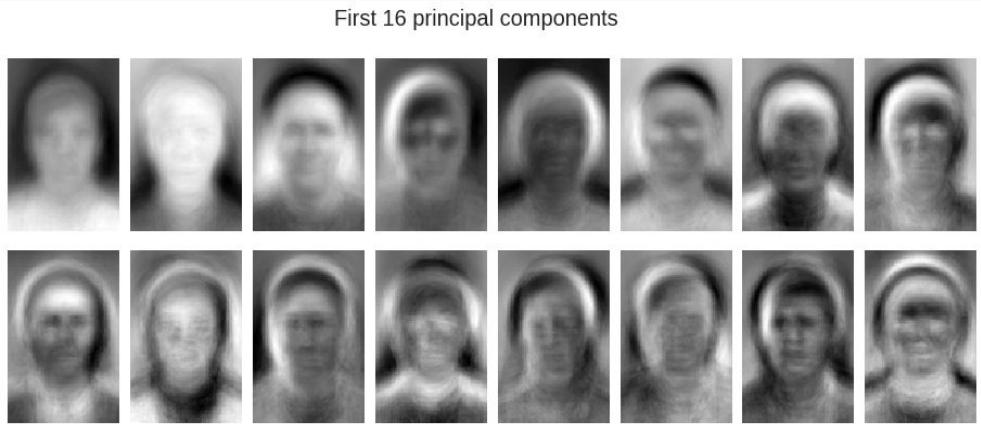
```
x_train -= x_train.mean(axis=0)

x_train /= np.std(x_train,axis=0)
```

Part IV. Data Preprocessing

- **Principal Component Analysis**
  - First 207 principal components explain over 95% of variance.



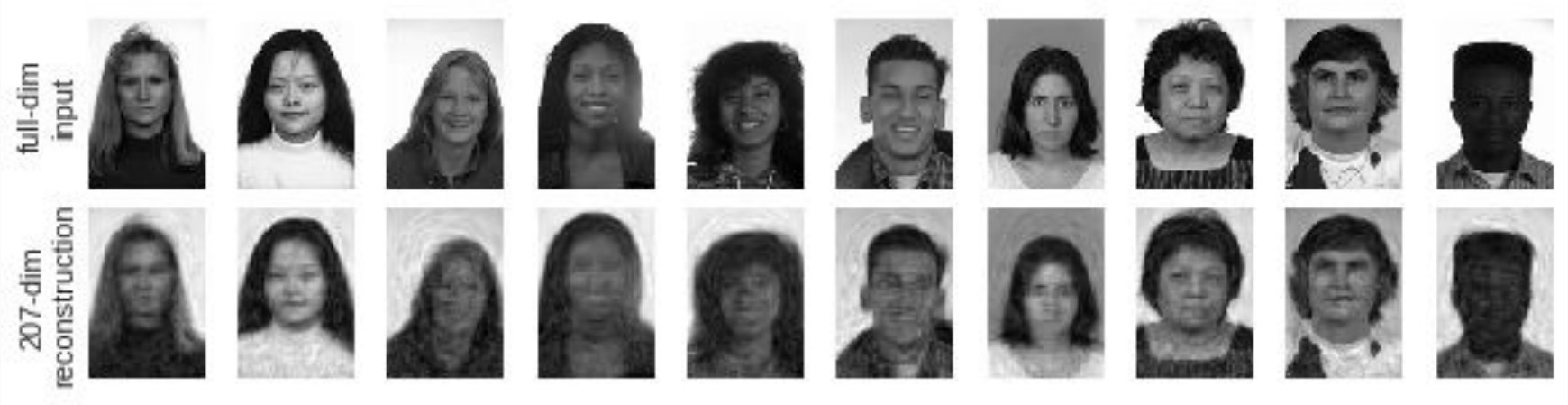


  - Visualize the images associated with the first several principal components

Part IV. Data Preprocessing

- **Principal Component Analysis(PCA)**

  From reconstruction: most of variance are maintained after dimension reduction.



Part V & VI. Data Analysis and Result Analysis

- **1. Support Vector Machines(SVM) - Model Selection & Result Analysis**

a. Hyperparameters:

C(controls the margin hardness): [1, 5, 10, 50]

Gamma(controls the size of radial basis function kernel): [0.0001, 0.0005, 0.001, 0.005]

b. Use the original training data with 8664 features:

C = 10, gamma = 0.0001

c. Use dimension-reduced data with 207 fundamental components (PCA):

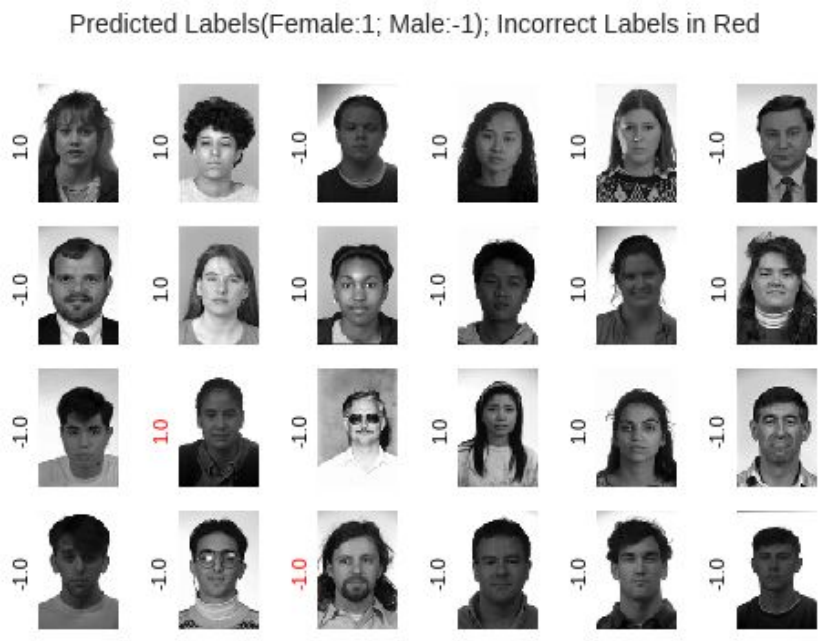
C = 5, gamma = 0.0001

d. Result:

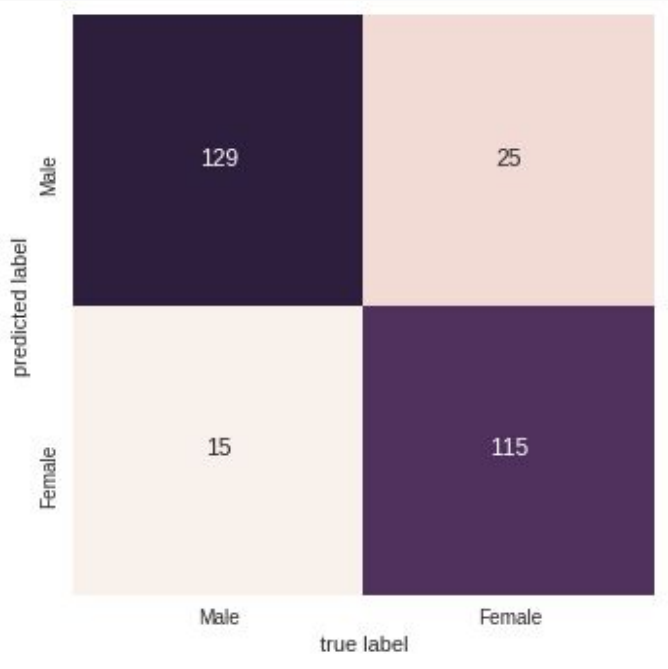
	Training error	CV error	Running time
Original Data	0.0004	0.1406	21.98
After feature selection	0.0174	0.1451	0.51
%change	+4250%	+3.2%	-97.7%

Part V & VI. Data Analysis and Result Analysis

- **1. Support Vector Machines(SVM) - Result Analysis**



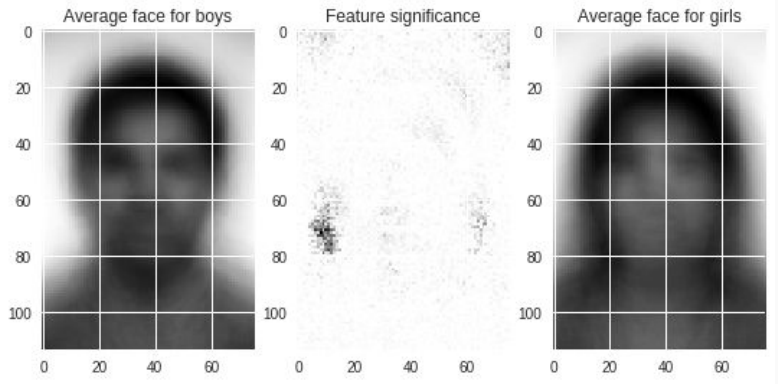
In the first several samples, most of them are classified correctly.

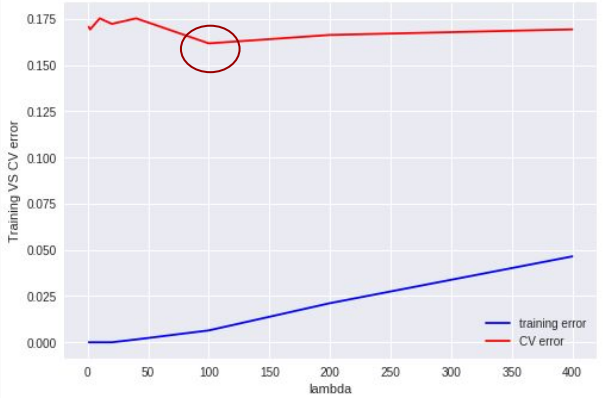


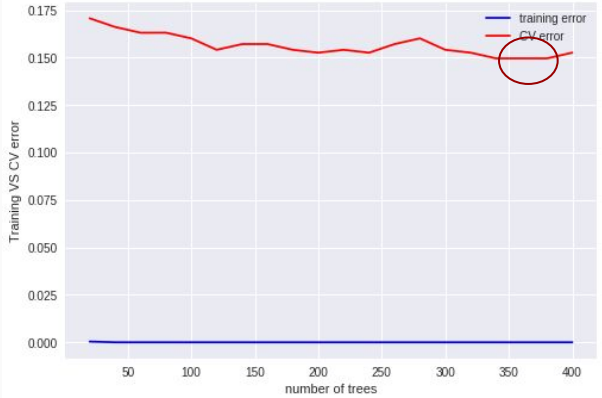
Confusion Matrix

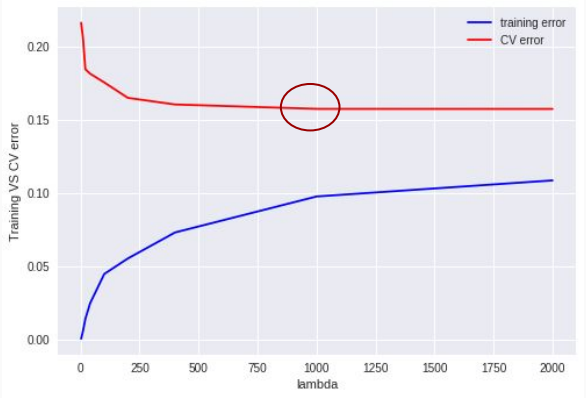
Part V & VI. Data & Result Analysis

- **2. Logistic regression - Model Selection**
  a. On original data:
     - select best regularization factor - 100
  b. Use random forest to select features:
     - 1400 pixels are being selected
  c. On data after feature selection:
     - Select best regularization factor - 20
  d. On data after PCA:
     - Select best regularization factor - 1000







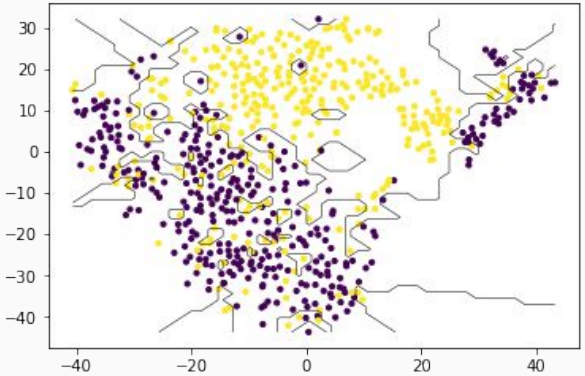


Part V & VI. Data & Result Analysis

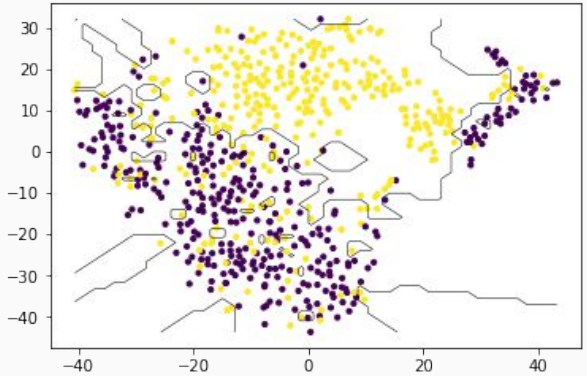
- **Logistic regression - Result analysis**
  a. Result:

	Training error	CV error	Running time
Original Data	0.0064	0.1617	7.8
After feature selection	0.0193	0.1466	0.69
After PCA	0.0816	0.1662	0.08

- **Logistic regression - Decision Boundary**



LR on original data

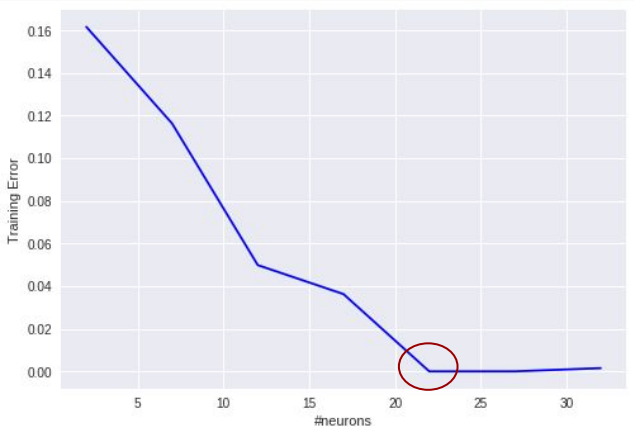


LR on data after feature selection

3. Neural Network -- Model Selection:

a. 3 main hyperparameters:
   - Number of layers
   - Number of neurons at each layer
   - Regularization coefficient

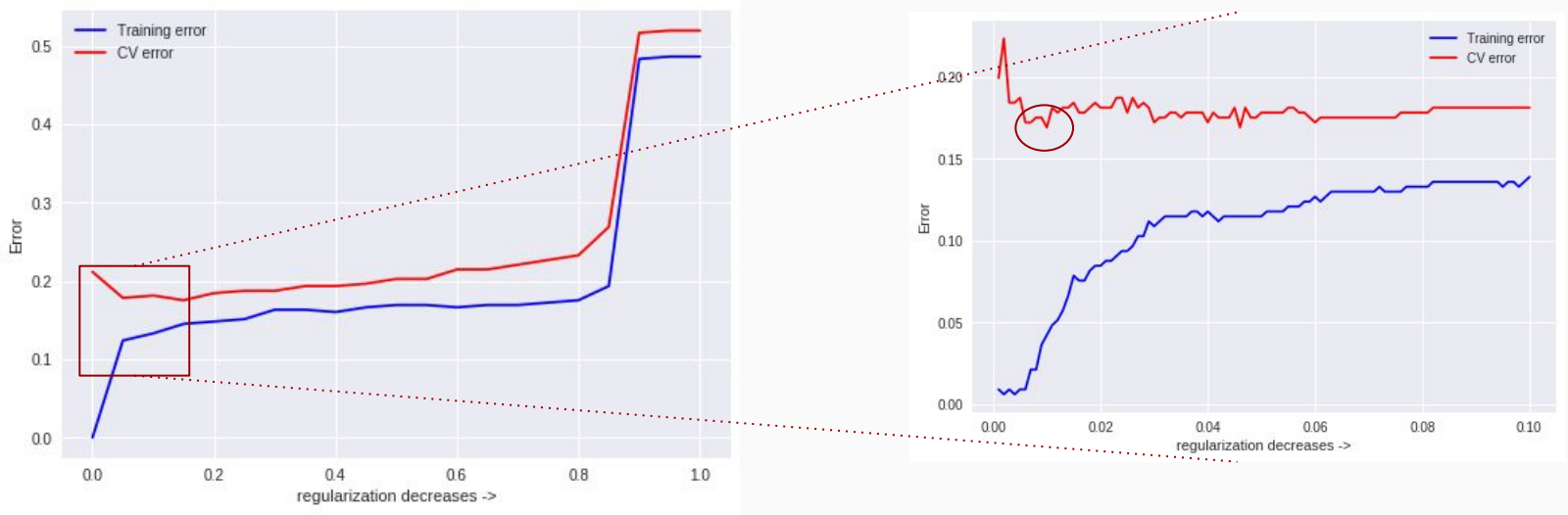
b. Fix lambda = 0, train a high-complexity model



At #neutron = 22, training error = 0, then we select value of regularization coefficient based on this high-complexity model.

Part V & VI. Data & Result Analysis

C. Fix model architecture (1 Layer with 22 neutrons, find regularization coefficient (x-axis value is reversed)



We select regularization coefficient to be 0.01

Part V & VI. Data & Result Analysis

- **3. Neural Network -- Result Analysis:**

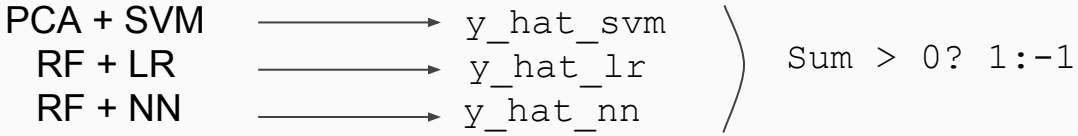
  Best NN model: 1 hidden layer with 22 neutrons with regularization coefficient = 0.01

	Training error	CV error	Running time
After feature selection	0.042	0.1690	208.19

Part V & VI. Data & Result Analysis

- **4. Simple Boosting on voting:**

We ensemble 3 algorithm together with simple voting method:

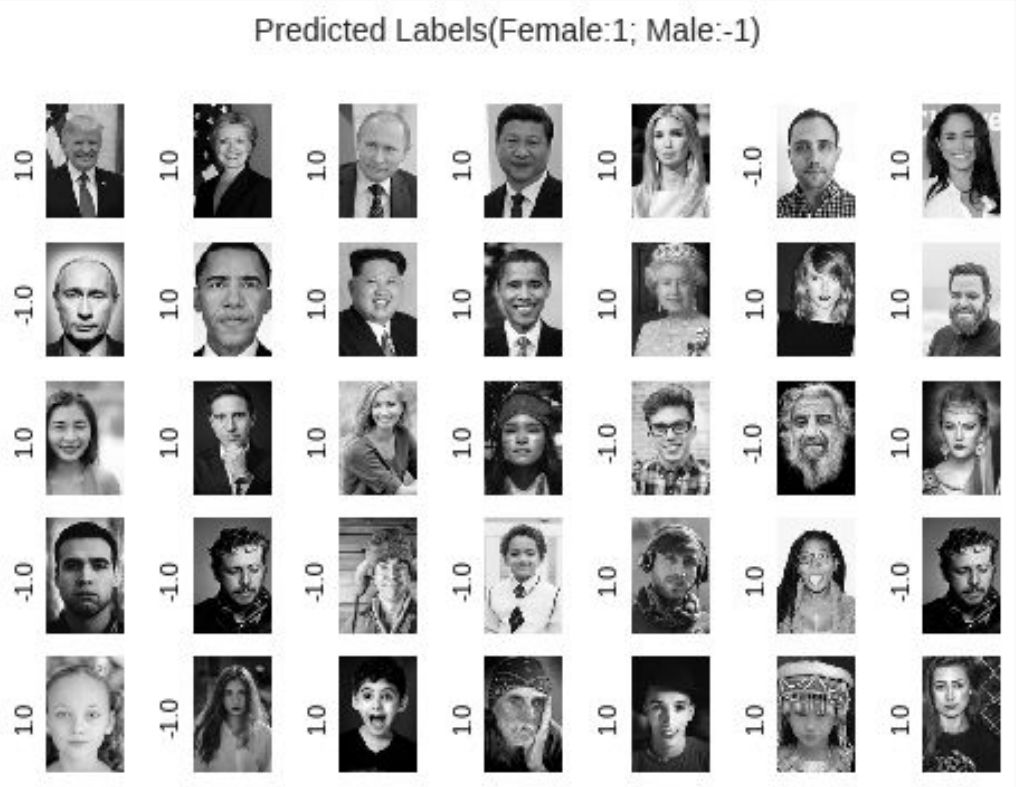


On testing dataset - predicting:

	Training error	CV error	Running time
Boosting	0.0	0.1690	0.8974

Part V & VI. Data & Result Analysis

- **Test on additional dataset**



Part VII. Future Work

I. Noise filtering using PCA

II. Be able to deal with different images

A. Not a front face:
   - be able to rotate the face or identify elements of face to classify gender

B. Face image not of the same size:
   - be able to adjust parameters according to different face size

C. Not a face image:
   - be able to identify the location of a face on an image

Thank you.

Github Repository: https://github.com/FengtongX/CE9010_project