M231 - Pattern Recognition and Machine Learning
STAT 231 / CS 276A - Project 3

DUCK-HA HWANG
UID#: 404589112

Face Social Trait and Political Election Analysis by SVM

1 Introduction and Objectives

This project is based on a paper by Jungseock Joo et al. in ICCV 2015. We consider a wide range of face social attributes including demographic, geometric and appearance facial features, and even some high-level perceptual dimensions (i.e. attractiveness, trustworthiness etc.). Furthermore, we exploit such facial attributes to analyze election outcomes and party affiliations (GOP vs DEM) of politicians. This study is motivated by prior behavior studies in psychology, which suggest that people judge others by facial appearance. Some evidence was also found in election and jury sentencing. Project 3 is an exercise for studying the social attributes of human faces using Support Vector Machines. Basically the goal is to:

1) train classifiers that can automatically infer the perceived face social traits from low-level features and
2) apply the model to analyze the outcomes of real-world political elections.

3 Tasks

3.1 Part I: Face Social Traits Classification (or Regression)

The goal of this task is to train binary SVMs (or SVRs) to predict the perceived traits (social attributes) from facial photographs. You can use the pre-computed facial keypoint locations and extract HoG (histogram of oriented gradient) features using the enclosed MATLAB function. You can further try your own favorite features.

[NOTE: We do not explicitly divide the image set into train/test sets. Therefore, you need to perform k-fold cross-validation and report the obtained accuracy.]

3.1.1 Part I-A: Classification by Landmarks

The first step of your assignment is to train 14 SVMs or SVRs only using the provided facial landmarks as features (no your own feature extraction step). Write a script which reads the annotation file and the landmark file and train 14 models -- one for each attribute dimension using the training examples. After training is done, you should apply the learned classifiers on the test examples and measure performance (classification accuracy) of the classifiers. Since the labels are imbalanced (different number of positive vs negative examples), you should report the average precisions.

[Things to Report: Training / testing errors and the "C" parameters you choose.]

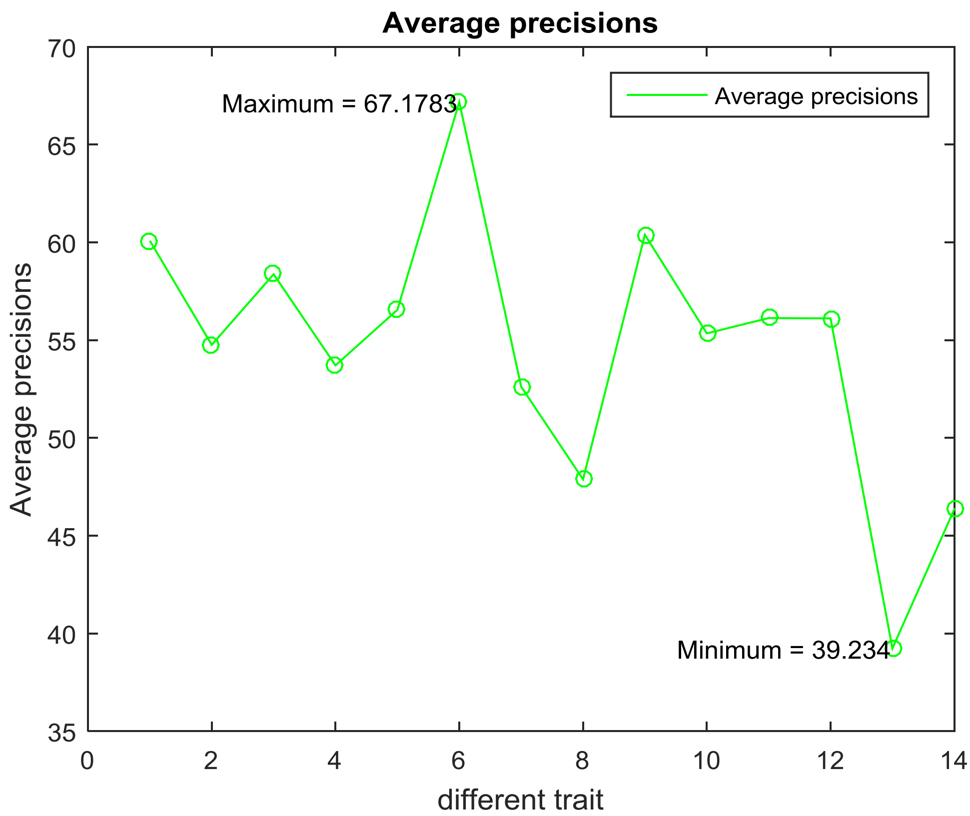
[NOTE: Through all experiments in this project, you need to measure both training error and testing error and report both in your report. When training SVM classifiers with LIBSVM or other libraries, you can specify a parameter ("C") to control the trade-off between classification accuracy and regularization. You should tune this parameter if you believe your classifiers are over-fitting.

Answer:

Using 5-fold cross-validation

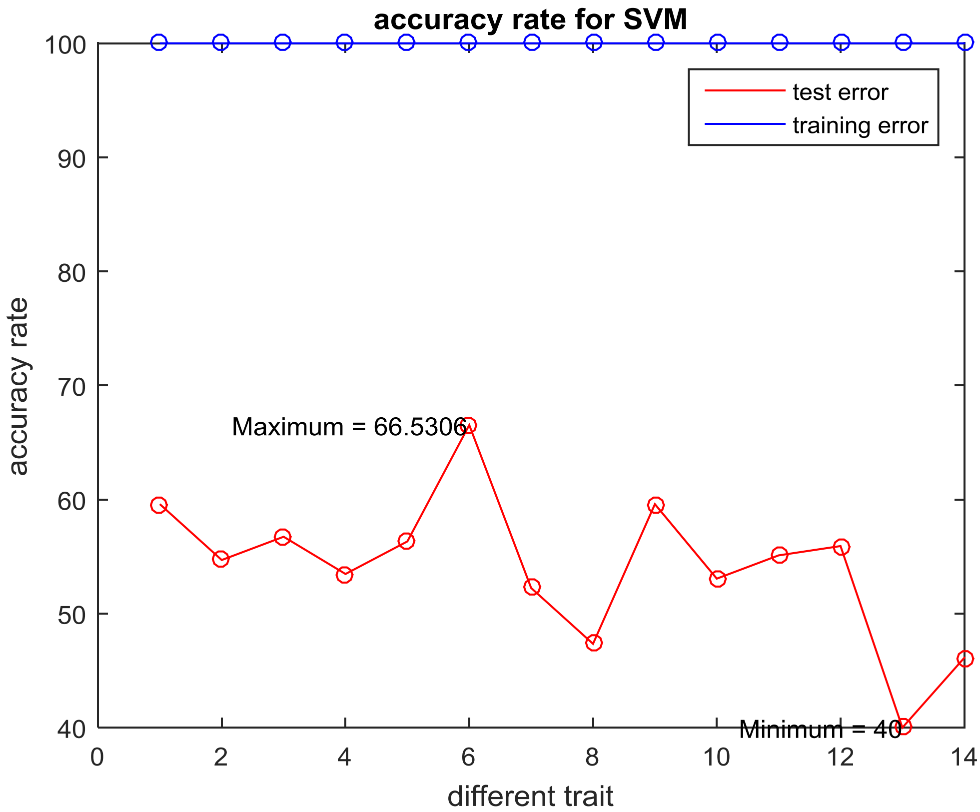
"C" parameters: 22	**Parameter: ('-s 1 -t 2 -c 22 –g 0.002')**

Average precisions graph:



Training / testing errors:

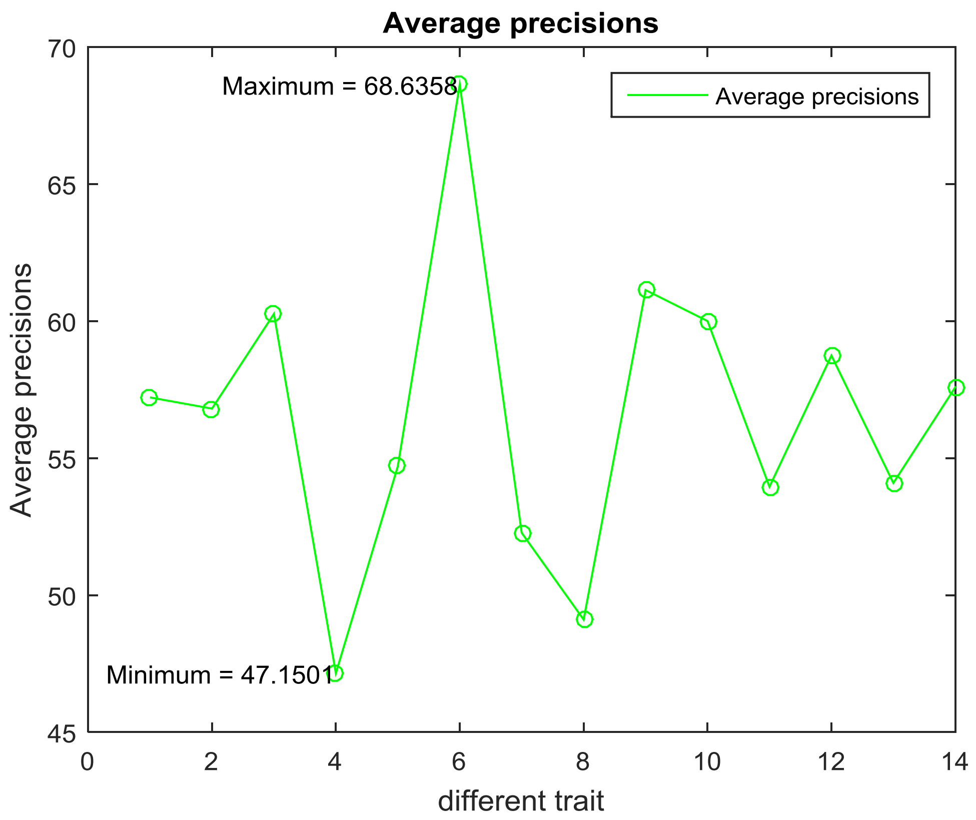
Training / testing accuracy:



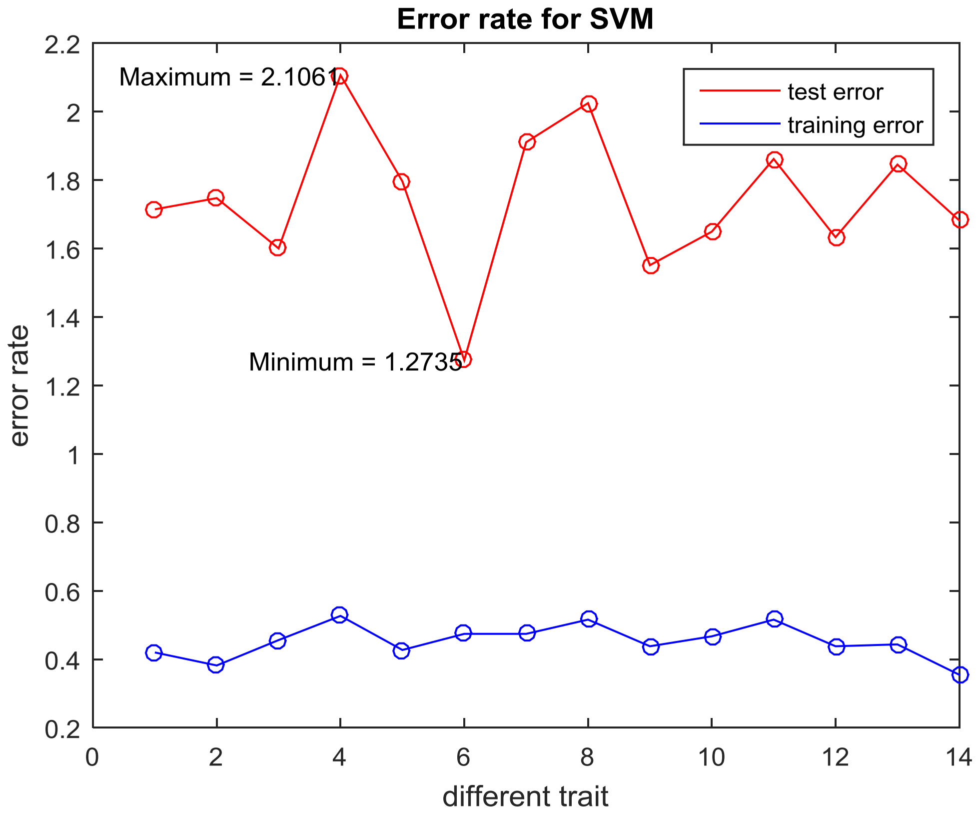
If I do normalize face landmarks then training error increase but testing error decrease.
Face landmark normalize:

"C" parameters: 22	**Parameter: ('-s 1 -t 2 -c 22 –g 0.002')**

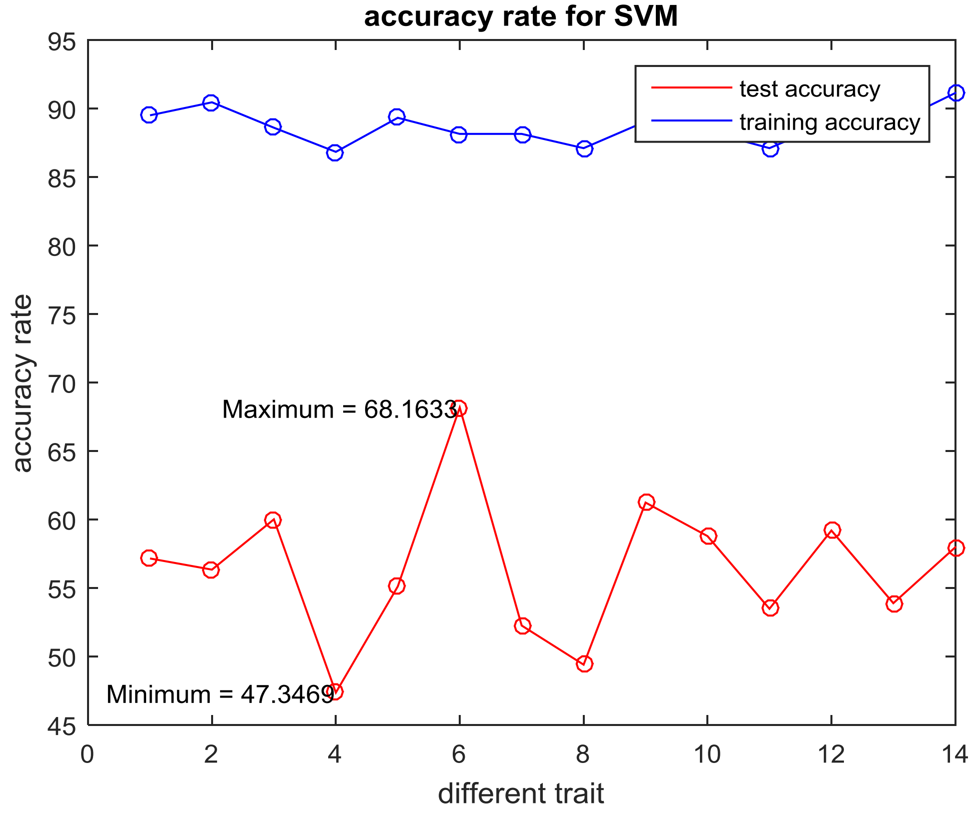
Average precisions graph:



Training / testing errors:



Training / testing accuracy:



3.1.2 Part I-B: Classification by Richer Features

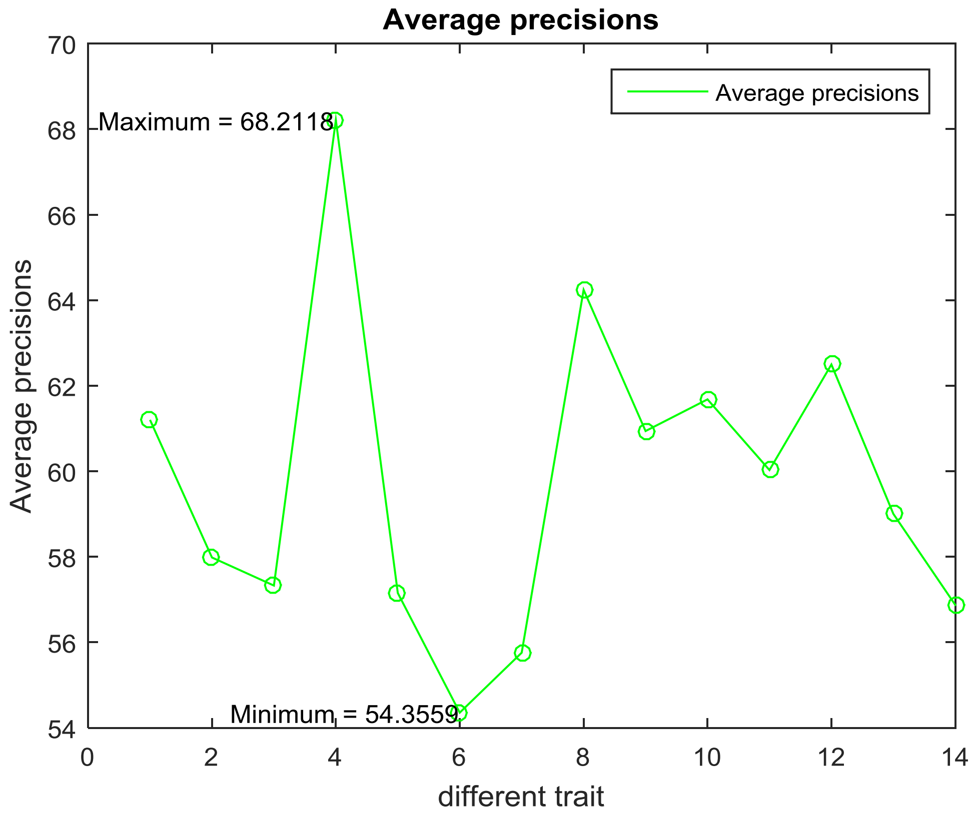
The next step is to extract richer visual features (appearance) from the images. Here, you should include the HoG (histogram of oriented gradient) features and can additionally choose whatever feature you want to try such as LBP (local binary pattern) or color histogram. Then repeat the earlier step to train and test SVM classifiers, but using augmented features: [landmark] and [new appearance feature]. You can simply concatenate two types of feature vectors into one. Compare the performance with the previous one.

[Things to Report: Names of the features you use Training / testing errors and the "C" parameters you choose.]

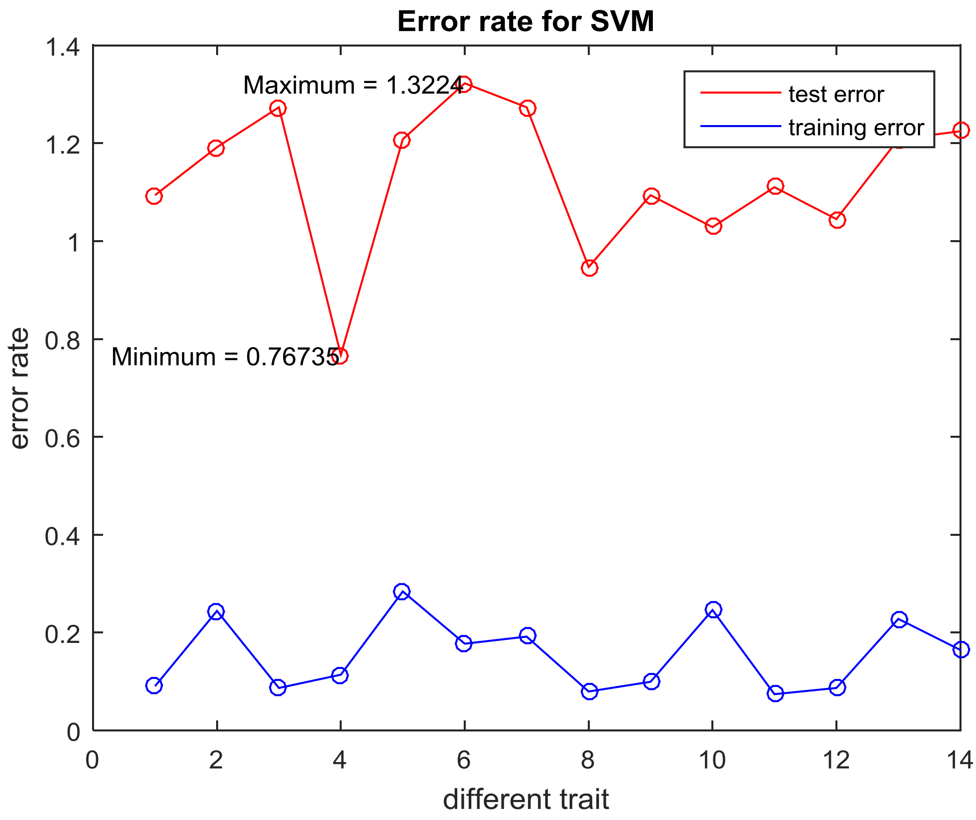
[Optional: You can further implement a joint feature representation such that you first locate a facial landmark and extract the appearance feature around the location. A similar treatment to this is the "**warping**" strategy that you used in the project 1.]

"C" parameters: 22	**Parameter: ('-s 1 -t 2 -c 100')**

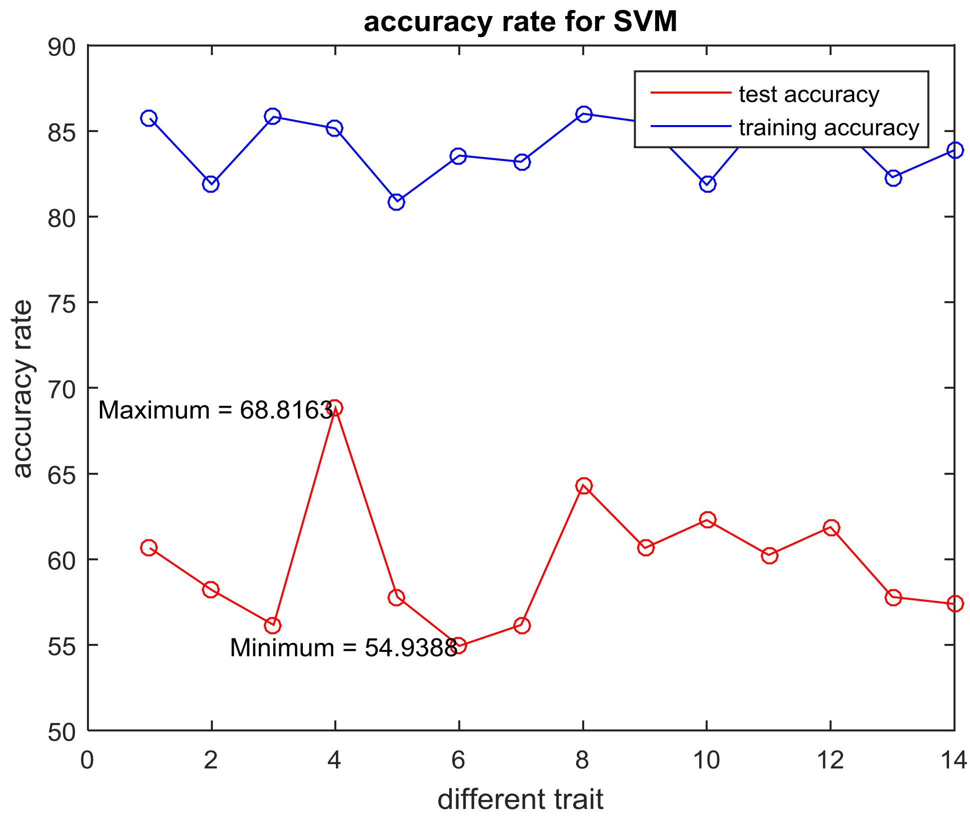
Average precisions graph:



Training / testing errors:

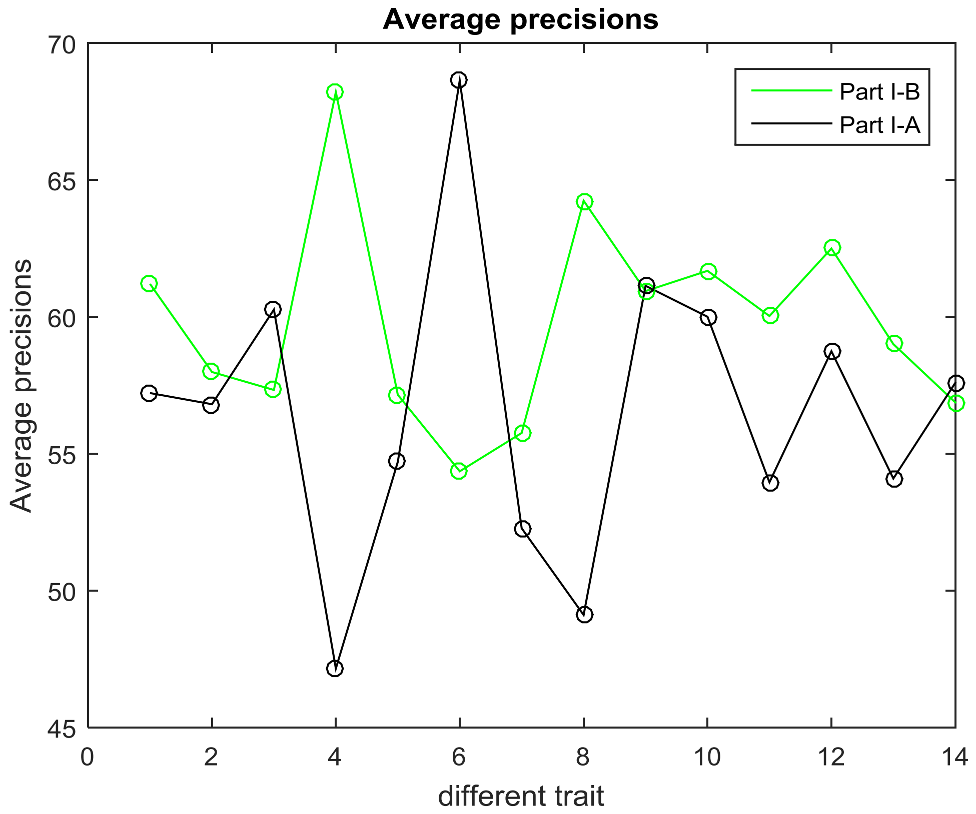


Training / testing accuracy:

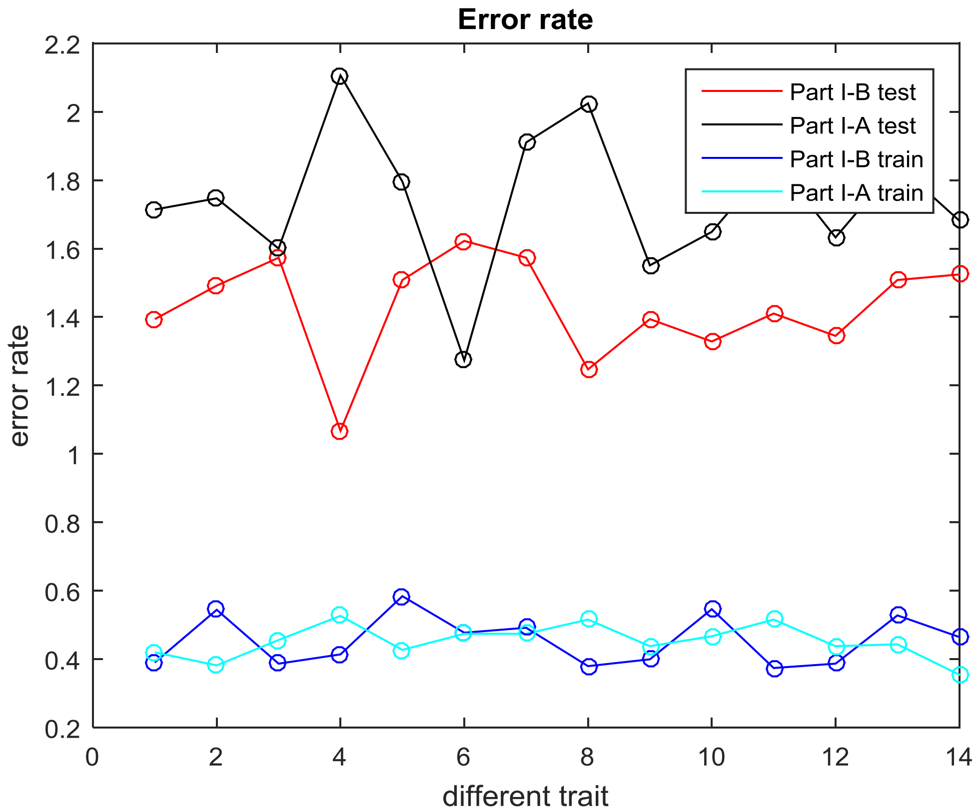


Compare the performance with the previous 3.1.1 result:

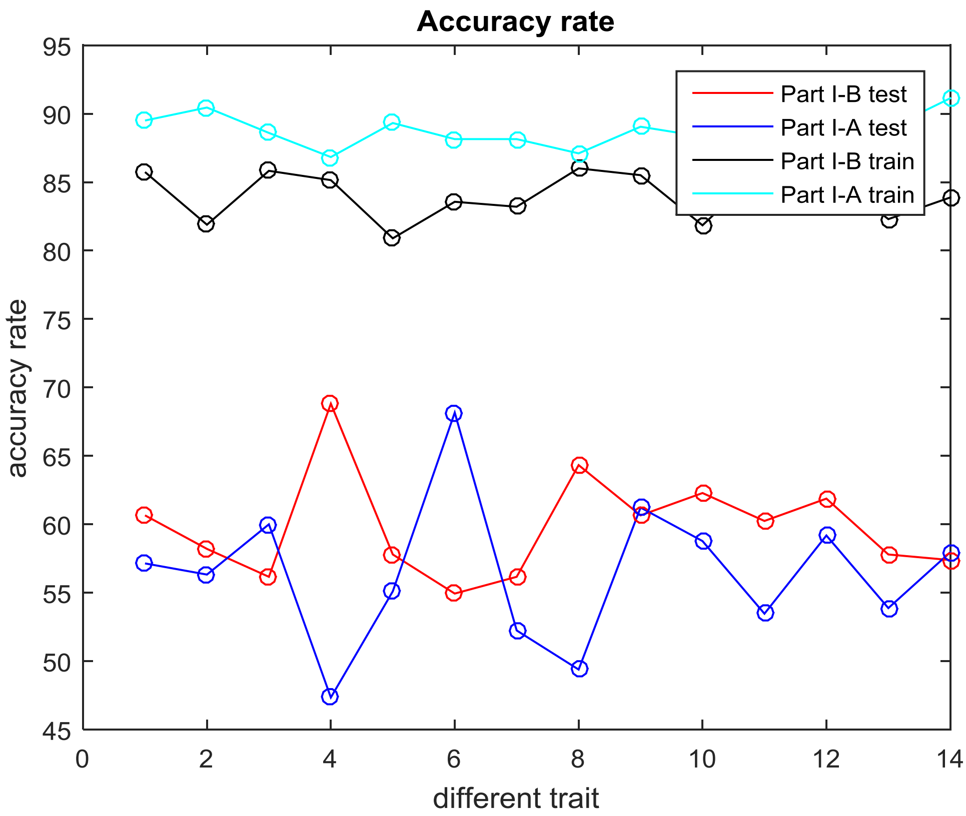
Average precisions graph:



Training / testing errors:



Training / testing accuracy:



3.2 Part II: Election Outcome Prediction

3.2.1 Part II-A: Direct Prediction by Rich Features

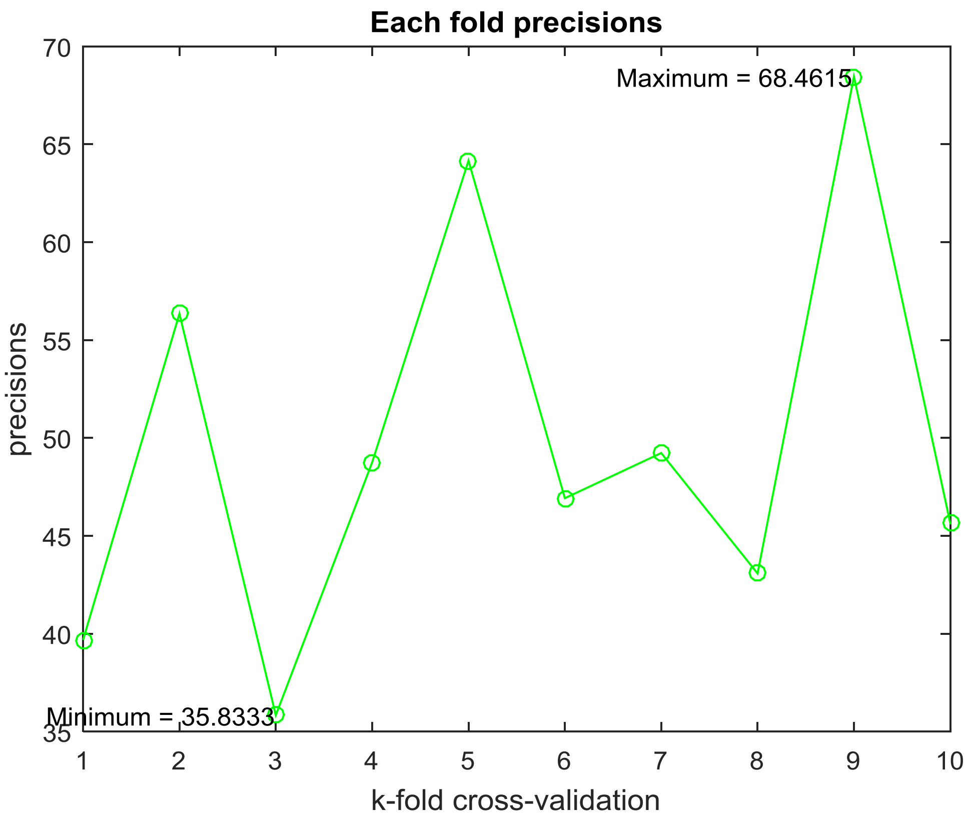
Using the same feature that you developed in the section 3.1.2, train a classifier to classify the election outcome.

[Things to Report: Training / testing errors and the "C" parameter you choose.]

[NOTE: We do not divide the second image set into a train and a test set. Perform k-fold or leave-one-out cross-validation and report the average accuracy. The point is to achieve an accuracy higher than chance.]

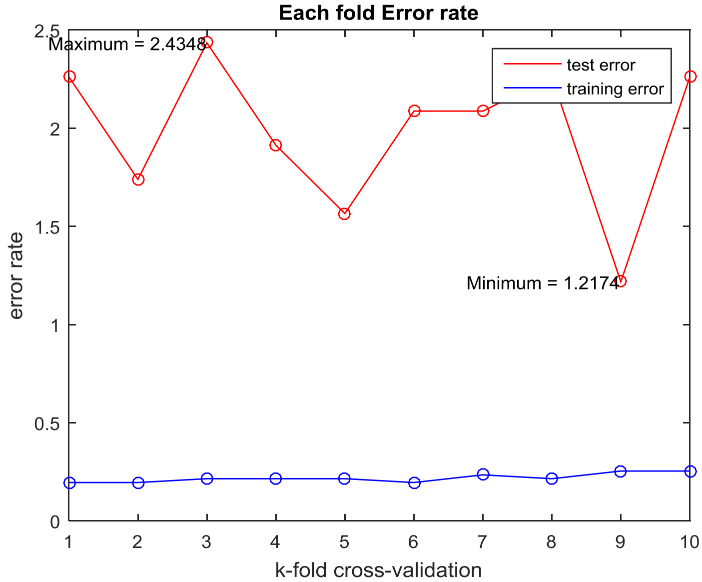
"C" parameters: 22	**Parameter: ('-s 1 -t 2 -c 100')**

Average precisions graph:



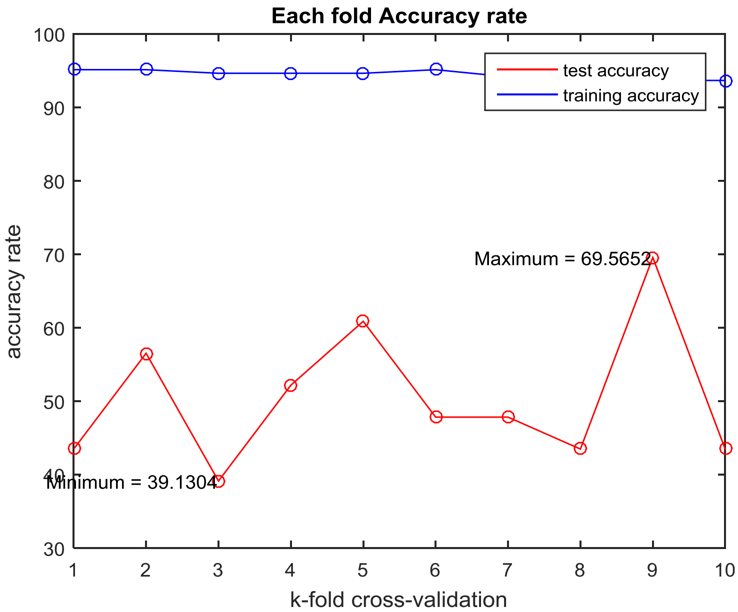
1-fold	2-fold	3-fold	4-fold	5-fold	Average
36.6825	56.3492	35.8333	48.7500	64.1667	49.8109
6-fold	7-fold	8-fold	9-fold	10-fold	
46.9231	49.2308	43.0769	68.4615	45.6349	

Training / testing errors:



1-fold	2-fold	3-fold	4-fold	5-fold	Average
2.2609	1.7391	2.4348	1.9130	1.5652	1.9826
6-fold	7-fold	8-fold	9-fold	10-fold	
2.0870	2.0870	2.2609	1.2174	2.2609	

Training / testing accuracy:



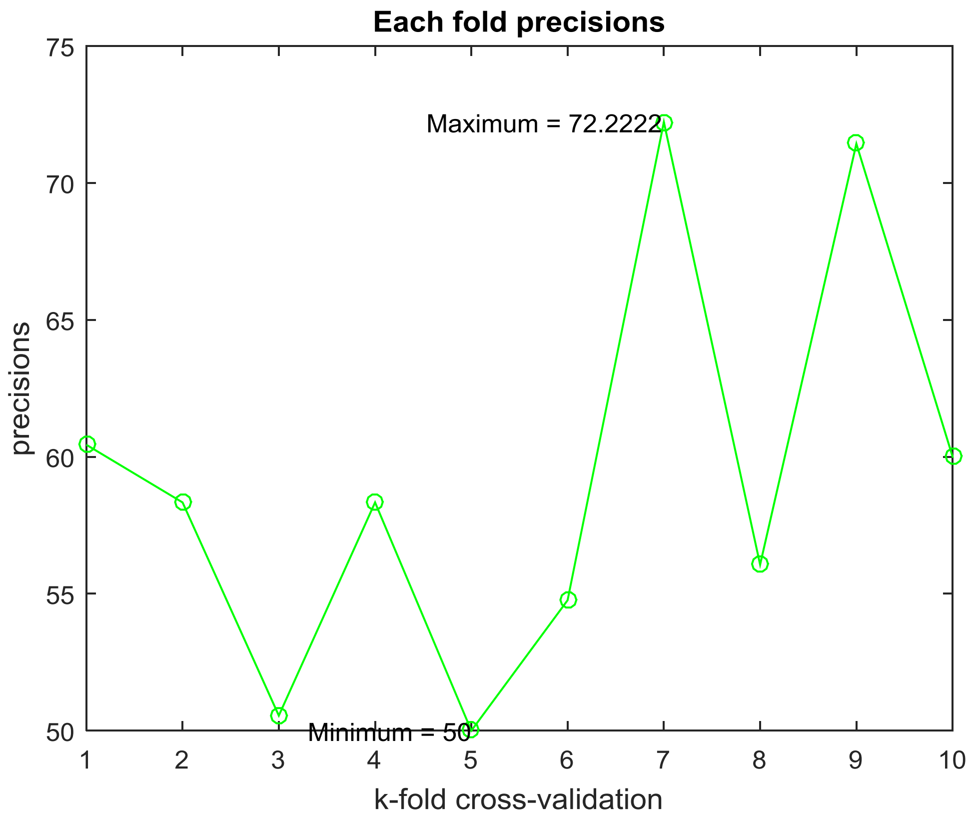
1-fold	2-fold	3-fold	4-fold	5-fold	Average
43.4783	56.5217	39.1304	52.739	60.8696	50.4348
6-fold	7-fold	8-fold	9-fold	10-fold	
47.8261	47.8261	43.4783	69.5652	43.4783	

3.2.2 Part II-B: Prediction by Face Social Traits

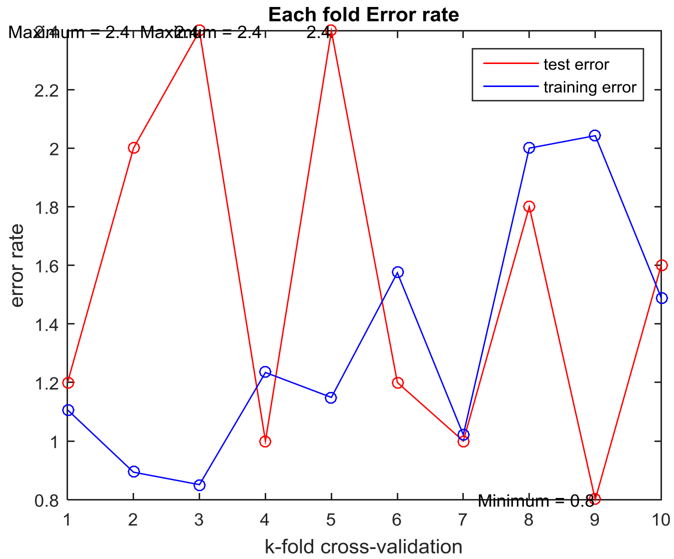
We finally consider a two-layer-model in which we first project each facial image in the 14-dimensional attribute space and classify the election outcome and the using the attribute-based representation. Specifically, you need to apply the classifiers that you trained in the section 3.1.2 to each politician's image and collect all the outputs of 14 classifiers (use real-valued confidence instead of label). Treat these outputs in 14 categories as a new feature vector that represents the image. Since each race comprises two candidates, a simple trick is to define a pair of politicians as one data point by subtracting a trait feature vector A from another vector B, and train a binary classifier: $\boldsymbol{F_{AB}} = \boldsymbol{F_A} - \boldsymbol{F_B}$. Do not include a bias term. Then you can again train SVM classifiers using these new feature vectors. Compare the result with direct prediction in 3.2.1.

[Things to Report: Training / testing errors, the "C" parameter you choose and comparison with 3.2.1.]

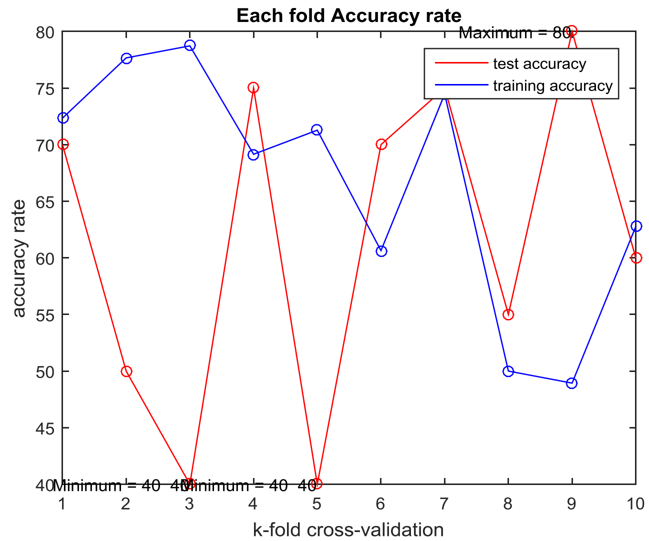
"C" parameters: 22	**Parameter: ('-s 1 -t 2 -c 100 –g 0.1')**



1-fold	2-fold	3-fold	4-fold	5-fold	Average
60.4396	58.3333	50.5495	58.3333	50	59.2129
6-fold	7-fold	8-fold	9-fold	10-fold	
54.7619	72.2222	56.0606	71.4286	60	



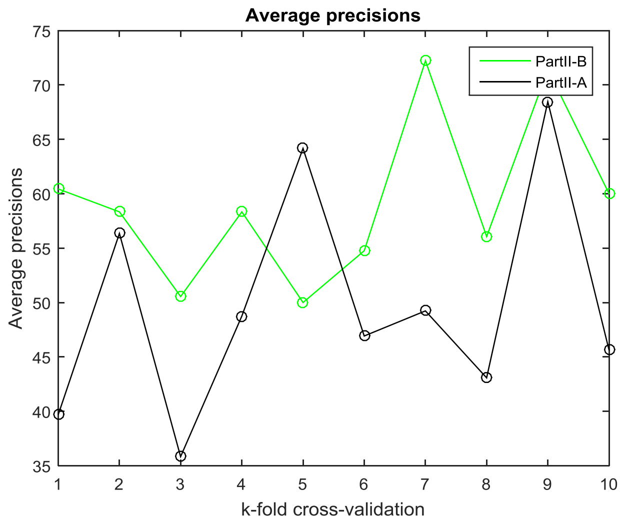
1-fold	2-fold	3-fold	4-fold	5-fold	Average
1.2	2	2.4	1	2.4	1.54
6-fold	7-fold	8-fold	9-fold	10-fold	
1.2	1	1.8	0.8	1.6	



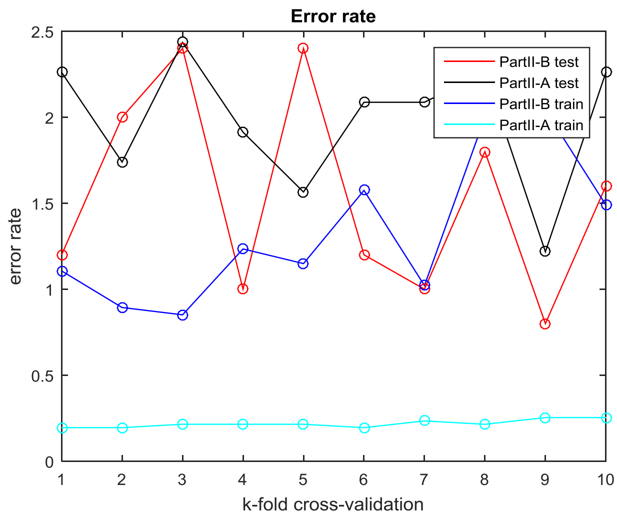
1-fold	2-fold	3-fold	4-fold	5-fold	Average
70	50	40	75	40	61.5
6-fold	7-fold	8-fold	9-fold	10-fold	
70	75	55	80	60	

Compare the performance with the previous 3.2.1 result:

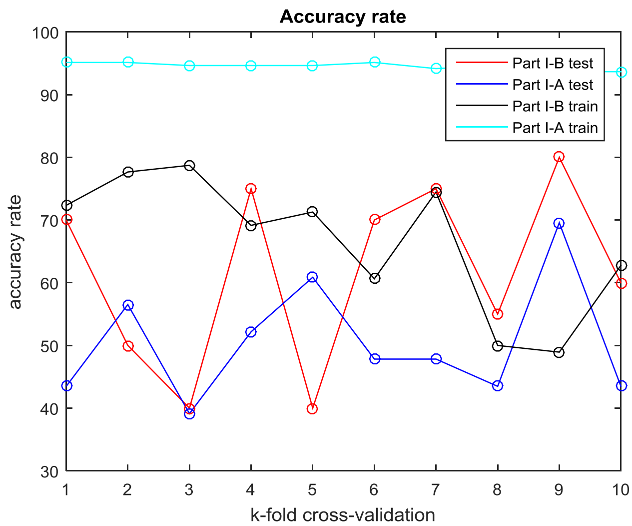
Average precisions graph:



Training / testing errors:



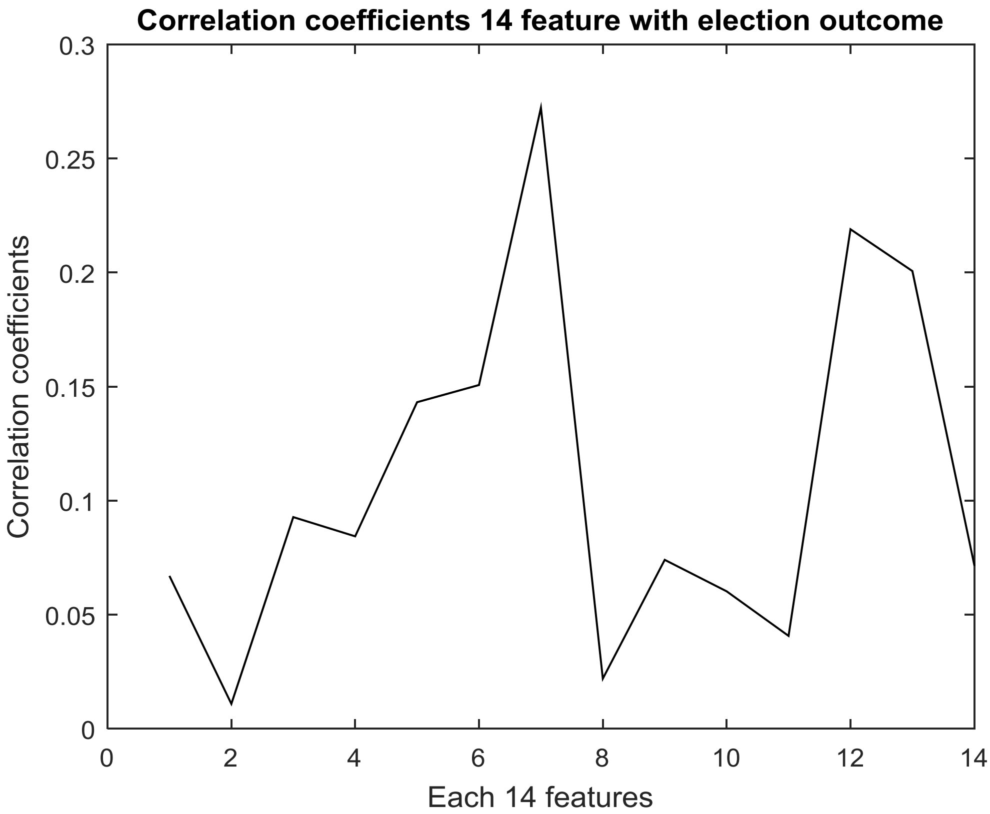
Training / testing accuracy:



3.2.3 Part II-C: Analysis of Results

At a minimum, show the correlations between the facial attributes and the election outcomes. What are the facial attributes that lead to the electoral success?

[Things to Report: Correlation coefficients of each of the facial attributes with the election outcomes.]



Correlations between the facial attributes and the election outcomes						
Old	Masculine	Baby-faced	Competent	Attractive	Energetic	Well-groomed
0.07	0.01	0.09	0.08	0.14	0.15	0.27
Intelligent	Honest	Generous	Trustworthy	Confident	Rich	Dominant
0.02	0.07	0.06	0.04	0.21	0.20	0.07

High Correlations: Well-groomed, Confident, and Rich

Low Correlations: Masculine, Intelligent, and Trustworthy