PROJECT REPORT
PROJECT 3

Anurag Pant
UID: 705085298

Introduction

In this project, we make use of binary SVMs (Support Vector Machines) and SVRs (Support Vector Regression) to try and accurately predict the perceived traits of candidates who are contesting in the governor and senator elections. We have been given facial photographs of the different candidates in these elections and the values of the following perceived traits: Old, Masculine, Baby-faced, Competent, Attractive, Energetic, Well-groomed, Intelligent, Honest, Generous, Trustworthy, Confident, Rich, Dominant. We also have the facial landmarks of each facial photograph. We use this feature data combined with the HoG features of each image (rich features) to train a model over the perceived traits data. This model can then correctly guess the traits of the testing images from the landmark and the hog features. These traits are in turn used to correctly predict the results of the election using another trained model.

1. Face Social Traits Classification (or Regression)

In this part of the project, we make use of the perceived traits from the facial photographs of the candidates. We have already been provided with the 14 facial trait values for each of the photographs. We have been provided with the landmarks for each image as well (160 landmarks per image). We have to make use of the landmarks and the hog features (rich features that can be extracted for each image) to train a model that will predict the traits for any candidate given the landmarks or/and hog features.

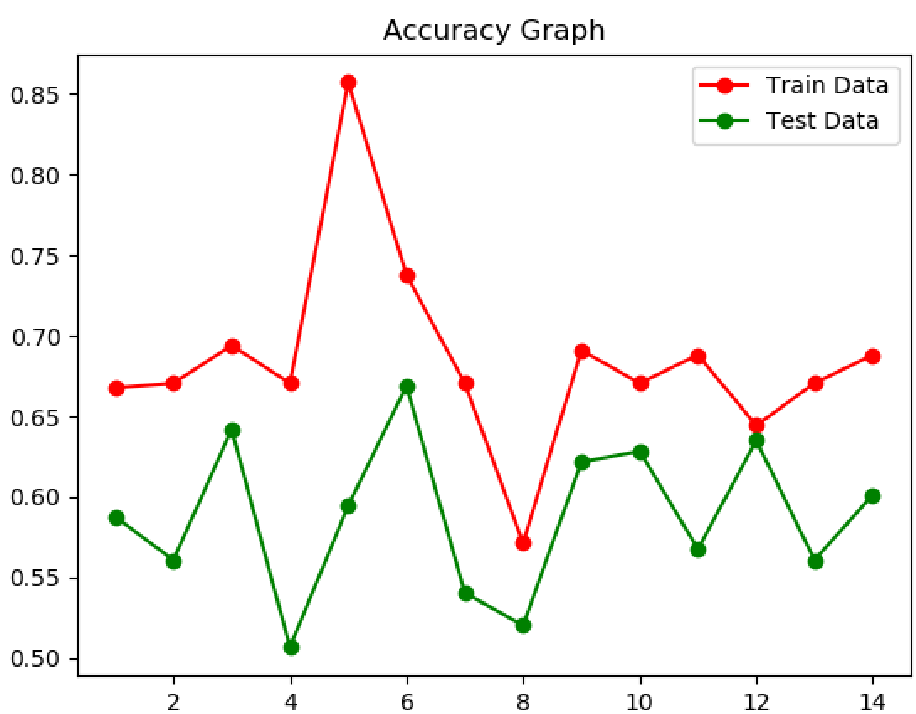
1.1. Classification by Landmarks

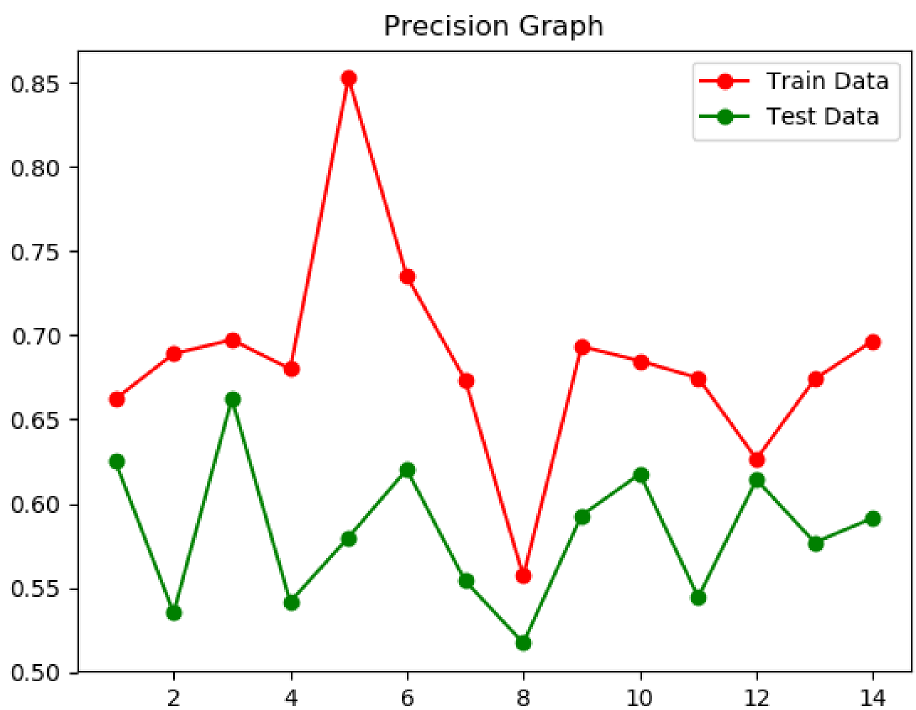
In this subpart, we make use of only the landmark features of each facial image to train our SVR (Support Vector Regression). To do this, we first have to find the label for all the social trait values for each image. We first divide our data into testing and training data. We find the labels by finding the threshold for each trait (by taking the mean of the social traits for the entirety of the training data). We then finding the training and testing labels. Now we train our model by taking the normalized landmark features as the training and testing data. We train by performing the k-fold cross-validation (with k = 10), we choose the parameters for the SVR (gamma, c and epsilon).

We train using the best parameters and return the predicted trait values for the testing data. We can then convert the predicted testing traits into labels using the thresholds that we had calculated earlier. These are then compared to the testing labels to find the accuracy and precision.

The training accuracy, training precision, testing accuracy and testing precision for each trait are given below:

Traits	Training Accuracy	Testing Accuracy	Training Precision	Testing Precision
Old	0.66	0.58	0.66	0.625
Masculine	0.67	0.56	0.68	0.53
Baby-faced	0.69	0.64	0.69	0.66
Competent	0.67	0.506	0.68	0.54
Attractive	0.85	0.59	0.85	0.58
Energetic	0.73	0.66	0.735	0.62
Well-groomed	0.67	0.54	0.67	0.55
Intelligent	0.57	0.52	0.55	0.51
Honest	0.69	0.62	0.69	0.59
Generous	0.67	0.62	0.68	0.61
Trustworthy	0.68	0.56	0.67	0.54
Confident	0.64	0.63	0.62	0.61
Rich	0.67	0.56	0.67	0.57
Dominant	0.68	0.601	0.69	0.59





The best parameters for each trait are given below:

Traits	C	Epsilon	Gamma
Old	8192.0	0.5	3.05 * e^-05
Masculine	0.5	0.125	0.03125
Baby-faced	32.0	0.125	0.001953125
Competent	0.03125	0.03125	0.125
Attractive	0.125	0.125	0.5
Energetic	0.5	0.125	0.03125
Well-groomed	0.03125	0.125	0.125
Intelligent	0.5	0.5	0.125
Honest	2048.0	0.03125	7.62 * e^-06
Generous	0.5	0.125	0.03125
Trustworthy	0.5	0.125	0.03125
Confident	0.125	0.125	0.03125
Rich	8192.0	0.125	7.62 * e^-06
Dominant	2048	0.125	3.05 * e^-05

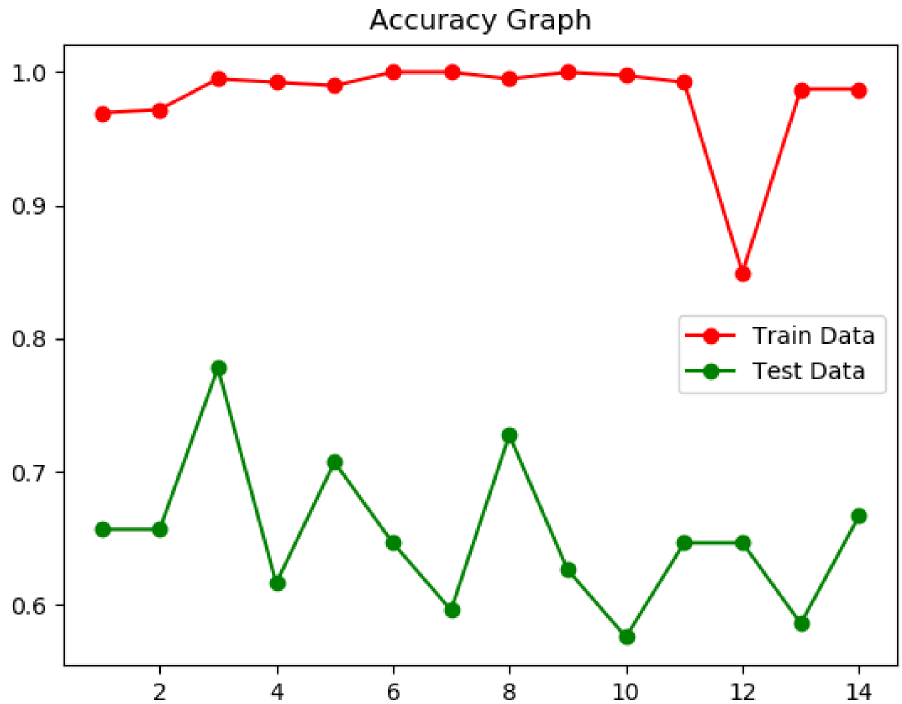
1.2. Classification by Rich Features

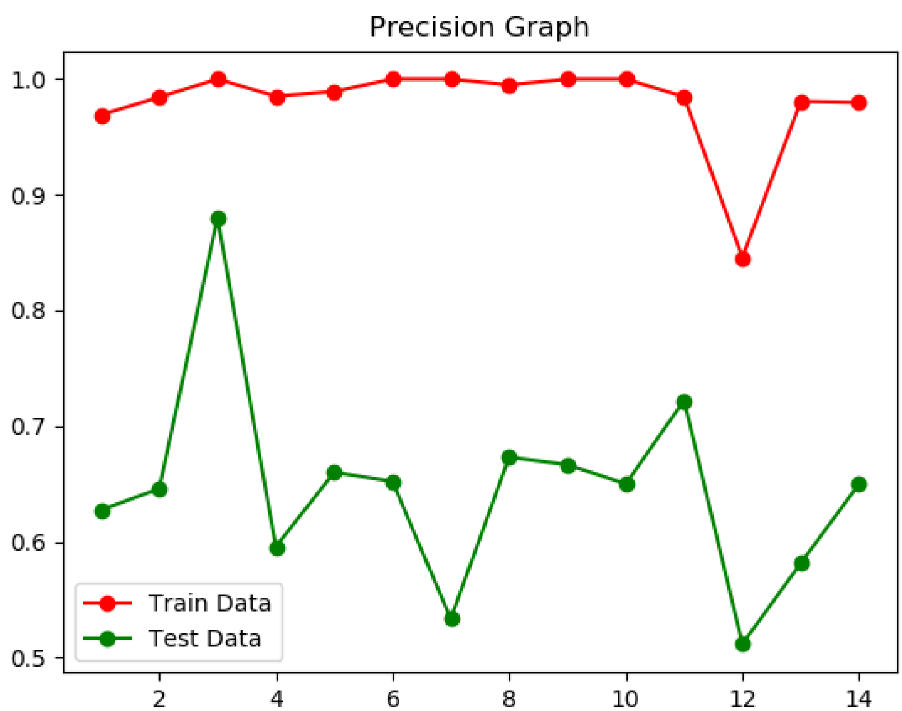
In this subpart, we make use of both the landmark features and the hog features to train the SVR model. We find the HoG (histogram of oriented gradient) features for each image by making use of the **hog function from skimage.feature library**. We concatenate the HoG features and the landmark features and normalize them along the columns. We then use these values and the labels, that we obtained using the thresholds (like in part 1.1.), for each image to perform cross-validation (k = 5). After finding the best parameters, we train our model. We then test over the testing data and find the values of the traits for the testing images. These trait values are converted to labels using the thresholds that we found for each trait. These are then compared to the testing labels and we find the accuracy and precision.

The only features I used for this part are **the landmark features that are provided for each image and the rich features that I extracted using the hog function**.

The training accuracy, training precision, testing accuracy and testing precision for each trait are given below:

Traits	Training Accuracy	Testing Accuracy	Training Precision	Testing Precision
Old	0.96	0.65	0.96	0.62
Masculine	0.97	0.65	0.98	0.64
Baby-faced	0.99	0.77	1.00	0.88
Competent	0.99	0.61	0.98	0.59
Attractive	0.98	0.707	0.98	0.66
Energetic	1.00	0.64	1.00	0.65
Well-groomed	1.00	0.59	1.00	0.53
Intelligent	0.99	0.72	0.995	0.67
Honest	1.00	0.62	1.00	0.66
Generous	0.99	0.57	1.00	0.65
Trustworthy	0.99	0.64	0.98	0.72
Confident	0.84	0.64	0.84	0.51
Rich	0.98	0.58	0.98	0.58
Dominant	0.98	0.66	0.97	0.65





The values clearly exhibit that the training and testing accuracies for part 1.2. are much better than the training and testing accuracies for part 1.1. This is because we are making use of the rich features along with the landmark features to train our model. The increased number of features allow for better training of the model and make room for increase in accuracy and precision.

The best parameters for each trait are given below:

Traits	C	Epsilon	Gamma
Old	8.0	0.03125	0.0001220703125
Masculine	8.0	0.03125	0.00048828125
Baby-faced	8.0	0.001953125	0.0001220703125
Competent	2.0	0.001953125	0.001953125
Attractive	8.0	0.0078125	0.00048828125
Energetic	512.0	0.001953125	7.69 * e^-06
Well-groomed	2.0	0.001953125	0.00048828125
Intelligent	2.0	0.001953125	0.00048828125
Honest	2.0	0.001953125	0.001953125
Generous	2.0	0.001953125	0.001953125
Trustworthy	8.0	0.0078125	0.00048828125
Confident	8.0	0.125	0.00048828125
Rich	2.0	0.0078125	0.001953125
Dominant	2.0	0.0078125	0.00048828125

2. <u>Election Outcome Prediction</u>

In this part, just like part 1, we make use of the normalized HoG and landmark features to train our model. However, instead of training our model to correctly estimate the traits of each image, we train our model to predict the outcome of the election given a pair of candidates. Since, we are making use of pairs of images that have to be compared against each other, instead of just predicting a trait for one image, we use the RankSVM model instead. The RankSVM works on minimizing the following equation:

$$\frac{1}{2}\|\vec{w}\|_2^2 + C\sum \xi_{i,j}^2$$

To implement a RankSVM, we need to find the difference in the features of the pairs using the features over which we wish to train our model. The dataset is divided into two sets of images: governors and senators. We train different model for the governors and the senators since, different features and traits may matter more in the senator and governor elections.

2.1. <u>Direct Prediction by Rich Features</u>

In this subpart, we make use of the difference (between each pair of candidates) of the concatenated HoG features and landmark features (after normalization) as the training data and take the labels as the results of the elections. Since, we have both the winner and loser in our datasets, we take half of the data as wins and the other half as losses. We do not make use of the traits that we had used in part 1 in this subpart of question

2. We use linearSVC model in this part of the question and we set the intercept to 0 to make it a RankSVM. We train 2 models, one for governor dataset and the other for the senator dataset.

We first perform k-fold cross-validation (k = 10) and find the best parameters. We then use these parameters to train our model. We then use the model on our testing data and find out the predicted election results. We compare this with the results of our testing data and find the accuracy.

The training accuracy and testing accuracy of governor dataset and senator dataset are given below:

Dataset	**Training Accuracy**	**Testing Accuracy**
Governor Dataset	1.00	0.66
Senator Dataset	1.00	0.58

The best parameters for both the models are given below:

Dataset	**C**
Governor Dataset	0.03125
Senator Dataset	0.03125

2.2. <u>Prediction by Face Social Traits</u>

In this subpart, we repeat what we did in subpart 2.2. However, this time instead of directly predicting the results of the election, we first predict the traits of each candidate in our election, using the trained model from 1.2. After getting the predicted traits for each candidate, we take out the difference of the traits (between each pair of candidates) and use that as the training data (after normalization). We use the results of the elections as the training label. We again take half of the data as wins and the other half as losses. Two different models are trained for governor and senator datasets.

We use linearSVC model to perform k-fold cross-validation (k = 10) and find the best parameters. We then train our models using these parameters and use the models on the testing data. We compare the predicted results with the results of our testing data and find the accuracy.

The training accuracy and testing accuracy of governor dataset and senator dataset are given below:

Dataset	**Training Accuracy**	**Testing Accuracy**
Governor Dataset	0.84	0.75
Senator Dataset	0.71	0.66

As is clearly evident from the training accuracies and testing accuracies, the traits are much more effective at predicting the results of the elections as compared to the HoG features and landmark features used in part 2.1. (despite the fact that the features in part 2.1. are much greater in number than the 14 trait features used in part 2.2.). This implies that the traits relate more to the way a candidate is perceived in elections and matter more than just the facial image and landmarks. They contain more information about the candidate and how he will fare in the elections as compared to his counterpart.

The best parameters for both the models are given below:

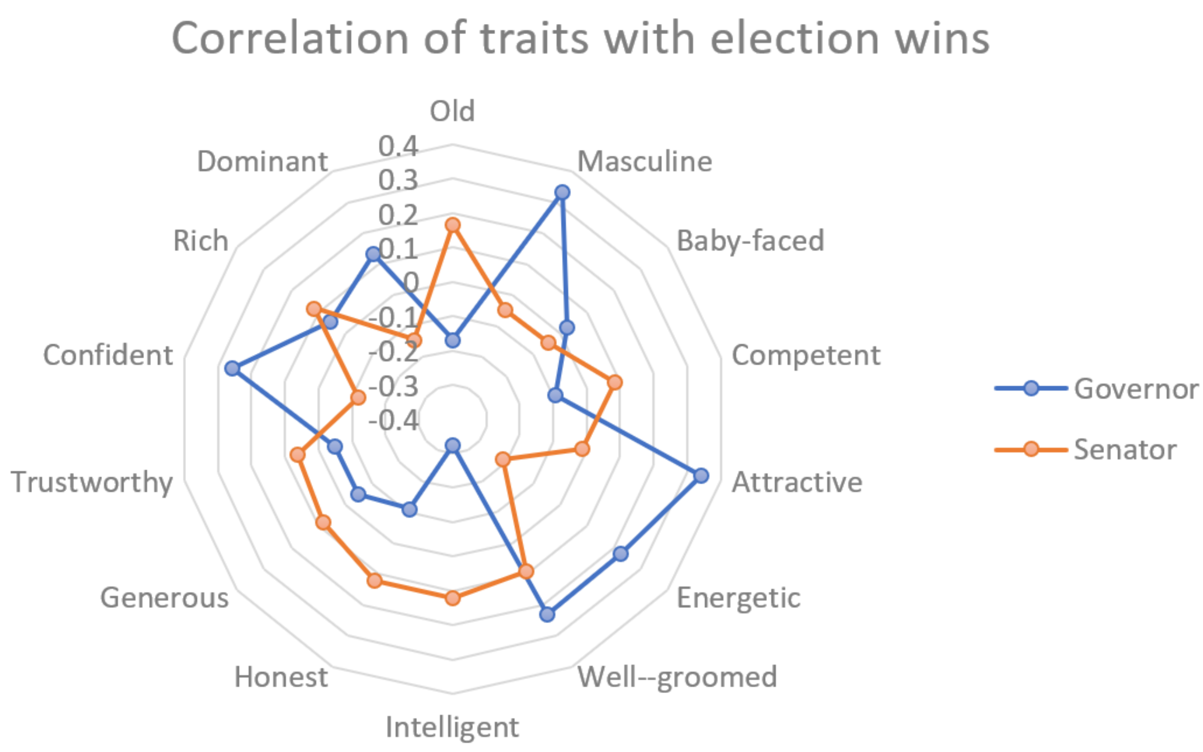
Dataset	C
Governor Dataset	32.0
Senator Dataset	8.0

2.3. Analysis of Results

In this subpart, we repeat what we did in subpart 2.2. However, instead of training the models after finding out the difference in the predicted traits between the different pairs of candidates, we directly make use of the results of the elections to find out the correlation between the traits and the election results.

The correlation coefficients for the governor dataset and the senator dataset are as follows:

Traits	Governor	Senator
Old	-0.16	0.16
Masculine	0.33	-0.049
Baby-faced	0.025	-0.041
Competent	-0.092	0.084
Attractive	0.34	-0.011
Energetic	0.22	-0.21
Well-groomed	0.23	0.093
Intelligent	-0.32	0.12
Honest	-0.109	0.12
Generous	-0.048	0.082
Trustworthy	-0.047	0.061
Confident	0.25	-0.12
Rich	0.053	0.11
Dominant	0.13	-0.14



According to the graph, being perceived as attractive, energetic, well-groomed, confident and masculine has a positive effect on the results of the governor elections. At the same time, being old and intelligent has a negative effect on the results of the governor elections.

According to the graph, being perceived as old, rich and intelligent has a positive effect on the results of the senator elections. At the same time, being energetic and dominant has a negative effect on the results of the senator elections.

The rest of the traits don't have much of an effect on the results of the respective elections.