The Eyes Have It

Max Curran, Marc Fadoul, Jeremy Gordon, and Liz Lee
Final Project for INFO 251: Applied Machine Learning, Fall 2017

Abstract: Eye tracking technology has the potential to offer more seamless interactions as input for a variety of software applications, but is there a cost to user privacy? We explore a dataset of eye tracking measurements and pupillometry while subjects viewed videos of actors making reciprocal eye contact, and investigate the dataset's predictive capabilities. Using a variety of machine learning models, we achieve significant improvements over a majority class baseline for predictions of observer gender, actor gender, and actor trustworthiness ratings. Our eye tracking-based classifiers also outperform personality-based classifiers on ratings of actor trustworthiness and attractiveness. In this analysis, however, eye tracking features were not predictive of other actor ratings or observer personality scores. Despite our use of one of the largest published eye tracking data sets available (N=405), we note that robust applications of machine learning rely on even larger sample sizes.

Motivation & Research Question

Eye tracking technology is increasingly embedded in consumer-facing products such as desktop or laptop computers (Tobii), virtual reality headsets (Pupil Labs, cognitive3D), and possibly soon in personal smartphones or tablets (Krafka et al. 2016) and smart eyewear (Behe et al. 2013, Calderone et al. 2015). Eye tracking has the potential to provide novel and/or easier forms of interaction; however, the analytics afforded by this new data type are not yet well-understood. thus the risk of inadvertent disclosure or breach of user privacy is difficult to assess and protect against.

Eye tracking with face stimuli has been used in many fields of research including social and behavioral psychology, human cognition and neuroscience, vision science and computer vision, and human-computer interaction. Scanning behavior when viewing faces has been indicated as individual-specific and temporally stable (Mehoudar, 2014). There has been some success predicting which emotion an observer is assessing in a face (Kanan, 2015), however only when training on observer-specific scan paths and not measures from across observers. The prediction of task being executed or mental state of an observer has been the subject of some debate, likely not possible using scan paths alone (Greene, 2012). Notable gender differences in viewing faces have been observed, such that female observers tend to focus more on a target's eyes (Hall, 2010) and males' on the nose and mouth when assessing emotion of a target (Vassallo, 2009). Cultural differences (east asian vs. western) in observer eye tracking behavior have also been found (Blais, 2008). Additionally, personality traits such as neuroticism have been associated with more time spent on a target's eyes (Perlman, 2009) and there is some evidence that sexuality can be predicted from eye behavior (Liebling et al. 2014)

Liebling and Preibusch draw a comparison between the ongoing proliferation of eye tracking and the ubiquity of webcams today, though importantly distinguish that in the case of webcams users are typically more aware of what they are disclosing and when (Liebling et al. 2014). By applying machine learning techniques to a dataset of eye tracking data in which observers view faces of actors, we investigate this concern with the question: What could be predicted about people and who they're viewing using only eye tracking data?

Data Source and Description

The data has been collected by Antoine Coutrot, a postdoctoral fellow in behavioral neuroscience, who conducted this study at the London Museum of Science with 405 participants (equal ratio of gender in the age range of 20-73 years) (data source: http://antoinecoutrot.magix.net/public/databases.html). He gathered basic demographic information upfront, such as their gender and age. There were three parts in the experiment that translated into three separate data sources: observer personality questionnaire, gaze task, and actor ratings by observers.

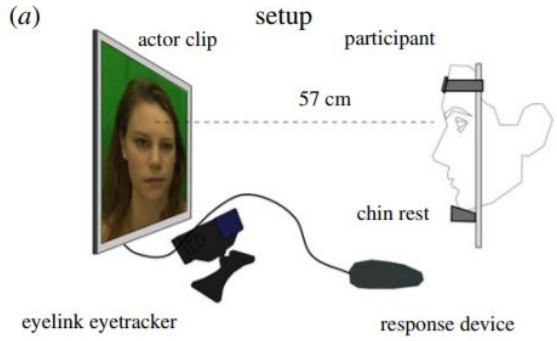
Participant Personality Questionnaire

Each participant completed the Big Five 10-item inventory (BFI-10), and were assigned a score on a scale of 1 to 10 on the personality traits of extraversion, conscientiousness, neuroticism, openness, and agreeableness.

Personality	**Min**	**Mean**	**Median**	**Max**	**SD**
Extraversion	2	5.1	5	10	1.7
Conscientiousness	2	4.8	5	9	1.5
Neuroticism	2	6.4	6	10	1.9
Openness	2	4.9	5	10	1.6
Agreeableness	2	5.2	5	10	1.5

Gaze Task

Next, observers sat across from a monitor with their head on a chin-rest where they looked at an actor gazing back from the monitor. Each observer was randomly paired with one of 8 actors (4 males and 4 females in the age range of 20-33 years) and sat through 35 different sessions of gazing at the same actor where each session lasted for a variable amount of time, from .1 to 10.3 seconds. During this time, eye-tracking data was collected at 250 Hz including the observer's pupil size and position of the eyes' target on the screen with X, Y coordinates.



Average across Sessions	**Min**	**Mean**	**Median**	**Max**	**SD**
Pupil size	278.6	3735.3	4507.1	9200.3	2616.6
Eye position (x, y)	(526, 458)	(635, 512)	(635, 512)	(719, 572)	(28, 14)

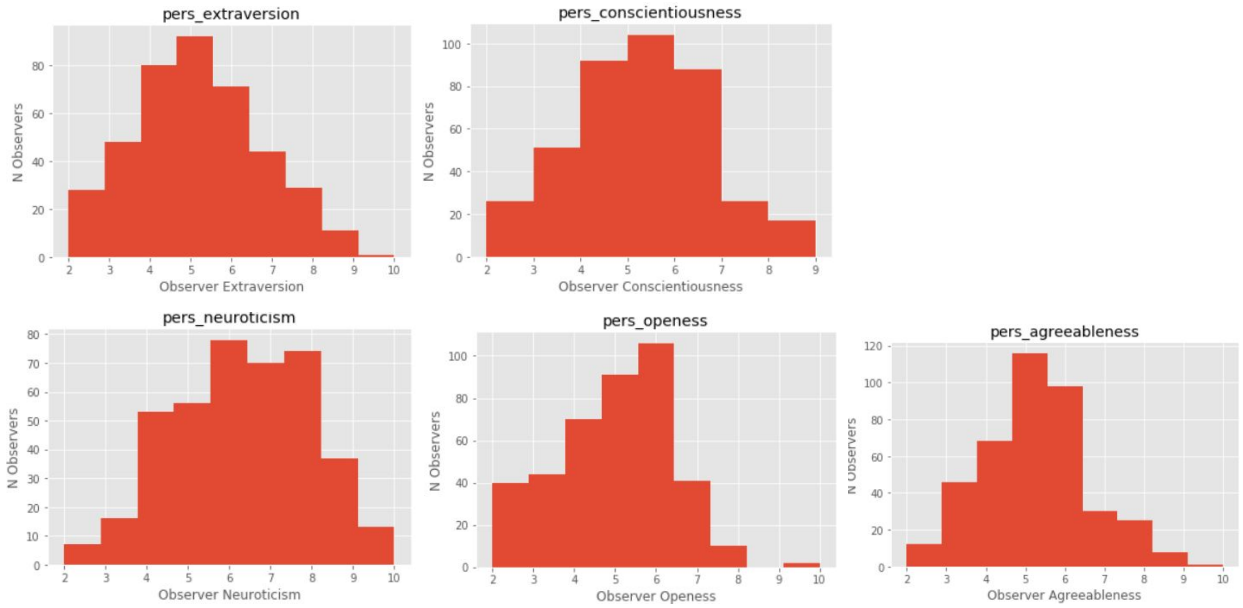
Actor Rating

After the gaze task, each observer rated the actor using a scale of 1 to 10 for the following descriptors: dominance, threat, attractiveness, and trustworthiness.

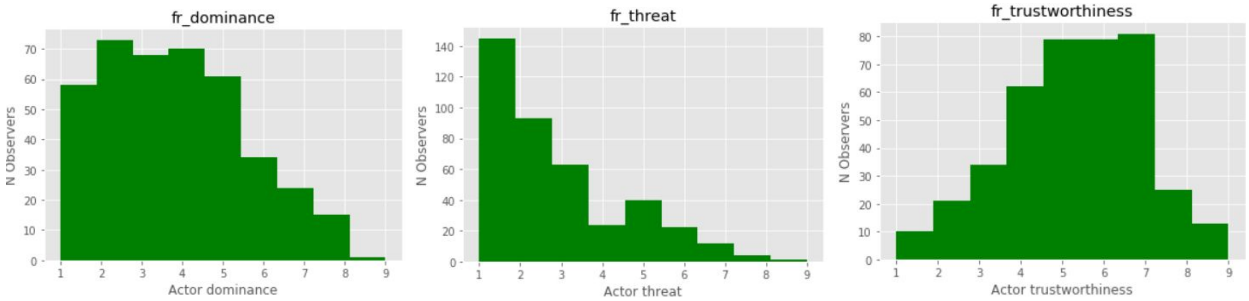
Descriptor	Min	Mean	Median	Max	SD
Dominance	1	3.7	4	9	1.9
Threat	1	2.7	2	9	1.8
Attractiveness	1	4.4	5	9	2.0
Trustworthiness	1	5.3	5	9	1.8

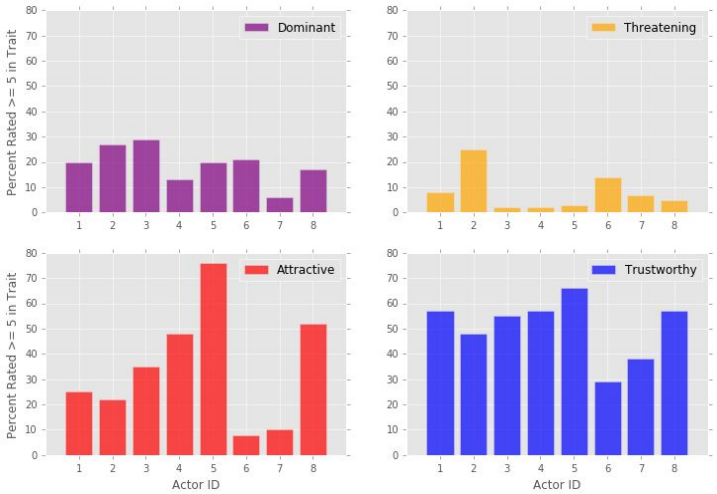
Exploratory Analysis

For our preliminary analyses, we looked at the distribution of both the observer's personality scores and actors' descriptors. Extraversion and conscientiousness roughly showed normal distributions, whereas the other three showed some directional skew.

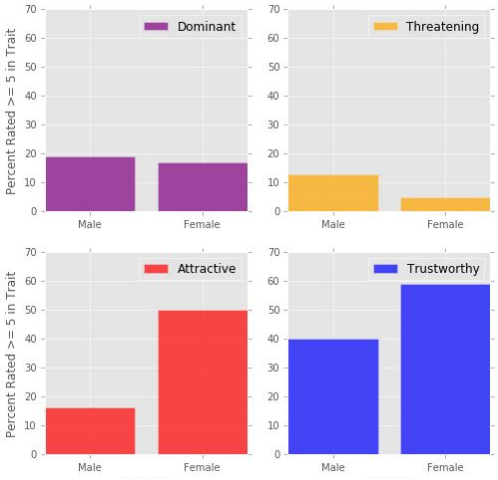


As for actors' descriptors, threat and dominance were highly skewed to the right while trustworthiness had a left skew, meaning observers rated the actors lower on the level of threat and dominance whereas they rated the actors higher on the level of trustworthiness.



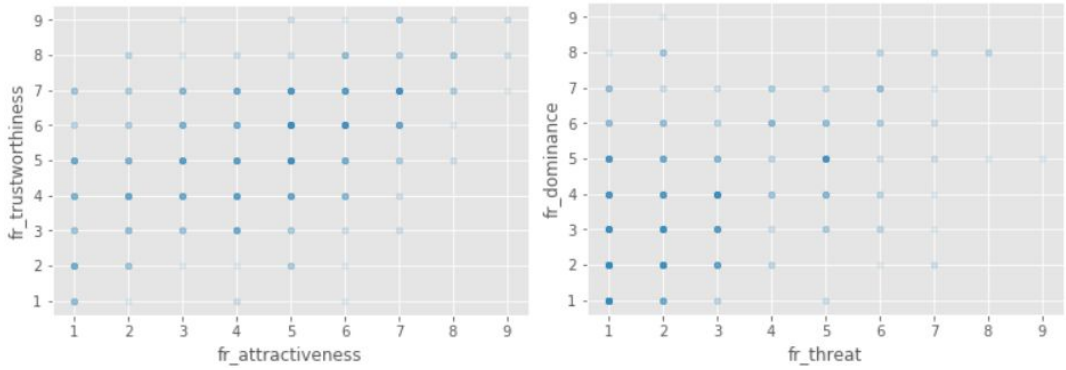


The four descriptors observers rated were different across different actors. For example, actor 5 was much more often rated at or above 5 on the attractiveness scale by observers, and actors 6 and 7 much less often than others.



Ratings were also associated with actor gender. Female actors were more often rated as attractive and trustworthy than male actors. Dominance was similarly linked to actor gender, in that male actors were rated threatening slightly more often.

Additionally, we looked at a correlation matrix of our dataset to find the Pearson coefficients between every pair of variables to see if we could detect any notable relationships. By finding these relationships, we can potentially consider variable importance when using models such as random forests. As a result we found some moderate level of correlation (between .3 and .49) between some variables: attractiveness and trustworthiness with r of +.49, threat and dominance with r of +.46, and attractiveness and actor's gender with r of +.37 (i.e. females were rated with higher attractiveness score).



The original authors of this dataset published two papers in 2016 in which they conducted their own set of analyses. The first of these papers (Bennetti 2016 et al.) explored the preferred gaze duration (PGD) for a large, diverse sample population and found an overall mean PGD of about 3.3 seconds, a positive linear relationship between age and PGD for male observers of female

actresses, and that PGD positively correlated with faster physiological arousal increase measured via pupil dilation; no relation was found with expected traits like gender, personality, or rated attractiveness. The second paper (Coutrot 2016 et al.) examined observers' eye scanning patterns when viewing actors' faces finding that both observer and actor gender influence gaze patterns, particularly that female observers engage in more exploratory patterns and female observers viewing female actresses tend to focus on the actress's left eye. Additionally, the authors used a set of features engineered from the eye tracking data to train a quadratic discriminant analysis (QDA) classifier to predict the gender of the observer (73.4% accuracy) and gender of both observer and the actor (51.2% accuracy).

Feature Extraction and Engineering

Basic features

Our targets for prediction (demographics, observer personality, actor ratings) were at the observer-level, so we created features at the observer level as well - collapsing across the many sessions for each actor-observer pair. We also engineered features that were time-independent, as some observers had longer sessions than others. The first features we created were simple statistics: mean and standard deviation of X and Y coordinates, and the ratio of these standard deviations. Standard deviation gives some approximation of scanning behavior in either horizontal (X) or vertical (Y), which we thought might have some predictive qualities. We also calculated the mean euclidean distance between readings, and the standard deviation of these distances. Mean and standard deviation measures were generally highly correlated, leading us to focus only on the standard deviations in our models as they tended to have more predictive power. These features combine eye motion in the X and Y directions, and can be thought of as more generalized measures of scanning behavior; observers with low standard deviations are more consistent in their eye motion between readings than those with high ones.

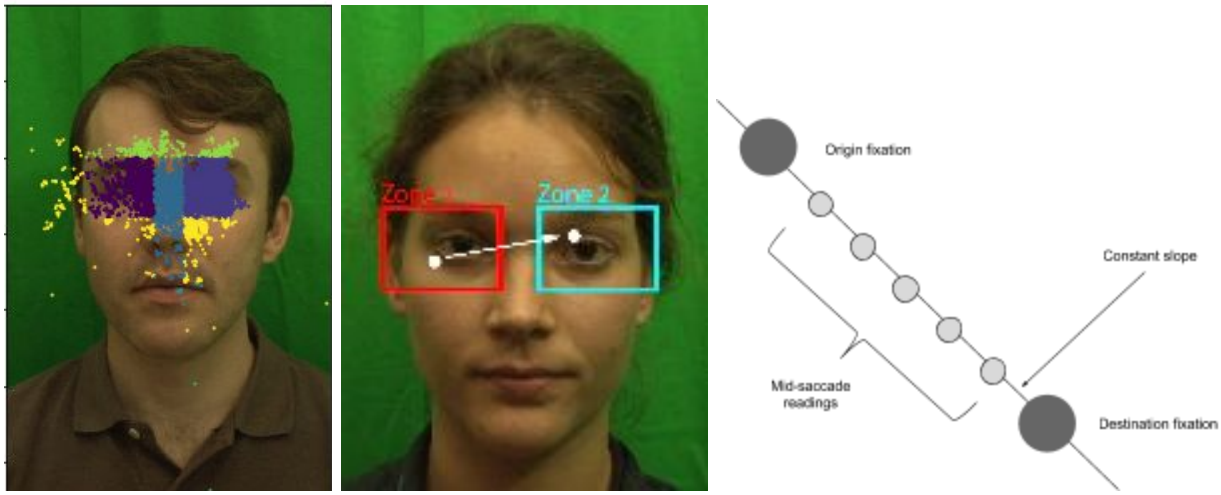
Saccades and zones

While some high level analysis can be performed on raw eye tracking data without an awareness of the target of a subject's gaze, adding context has the potential to significantly increase both the predictive power and interpretability of this kind of data. In the case of our target dataset, first step was to label each eye tracking record with the feature of the actor's face that the coordinates aligned with. In a prior eye tracking research project (Singh et al. 2016), dwell time on specific facial areas of interest (AOI) was seen to differ dependent on features of the target face.

The provided dataset included a sample still-frame for each actor, and these were used to manually define zones for each of six areas of interest: left eye, right eye, nose, mouth, forehead, below chin. All eye tracking readings not falling into one of these zones were labeled as zone 7. It was necessary to map the raw eye tracking data onto the facial coordinate system using provided face registration coordinates, before applying the proper zone to each reading.

The applied zone labels (see figure below) allowed us to create a distribution for each subject-actor session (i.e. the probability of a given eye tracked reading to fall into each of the seven zones).

In order to add temporal context to eye tracking data, it is necessary to move beyond a simple coordinate distribution and look at the level of saccades, which are a rapid eye movement between two fixation points, which typically occur subconsciously 4-5 times per second.



Zones applied to actor still frame (left). Illustration of zone 1 to zone 2 saccade (middle). Extracting saccades from raw eye tracking reading sequences (right)

Due to the high recording rate of the eye tracking system (250 Hz downsampled to 60 Hz), many points were recorded for any single saccade. To extract a saccade level dataset, the slope between all adjacent tracking points was calculated, and sequences of near identical slopes were identified. The edges of this sequence (first and last point) were interpreted as saccade origin and destination, and Euclidean saccade distance was also added.

This saccade level dataset allowed the addition of 49 features ($7!/(7-2)! + 7$), the distribution in each session of all permutations of origin zone and destination zone, including same zone saccades.

FFT and pupillometry data

In signal processing, it often comes in handy to extract features from the frequency space: filters allow the extraction of a relevant variation range, and peak frequencies can be used for characterization (such as voice pitch). This motivated an attempt to featurize the Fourier transform of the eye position (x and y) and pupil diameter.

We split the power spectrum of these three signals into buckets. Given the small size of the dataset, featurizing the whole spectrogram results in strong overfitting, even with a small

number of buckets. A more parsimonious strategy is to keep the frequency and intensity of the main peaks. Unfortunately, they did not significantly increase our prediction rates for gender, actor ratings, or personality scores.

There are three ways to interpret this lack of improvement:

- Our data does not have cyclical patterns. This is not completely true as the spectra show significant peaks.
- Our dataset is too small to train a relevant representation of the spectrum.
- The spectrogram does not contain relevant information for our prediction task. Note that some signals may be irrelevant for one task but still predictive for another task, just like a fingerprint is useless to predict personality traits but perfect to identify individuals.

Models for Analysis

Simple analysis of the dataset with a bi-dimensional PCA or LDA gives us two intuitions:

- The dataset does not seem linearly separable. This motivates the use of non-linear models.
- These techniques resulted in highly tangled labels. Even if the 2D projection only represents about 30% of the variance, it suggests that the classification will not be perfect, whichever model we choose.

The hyper-parameters of every model are cross-validated with a grid search. The range of parameters to be explored are hand defined. Keeping in mind the small size of our dataset, we limited the number of parameters (e.g. tree depth). For instance, we try several combinations of layers for the 2 layer neural network, but we limit the total number of neurons to 6 or fewer.

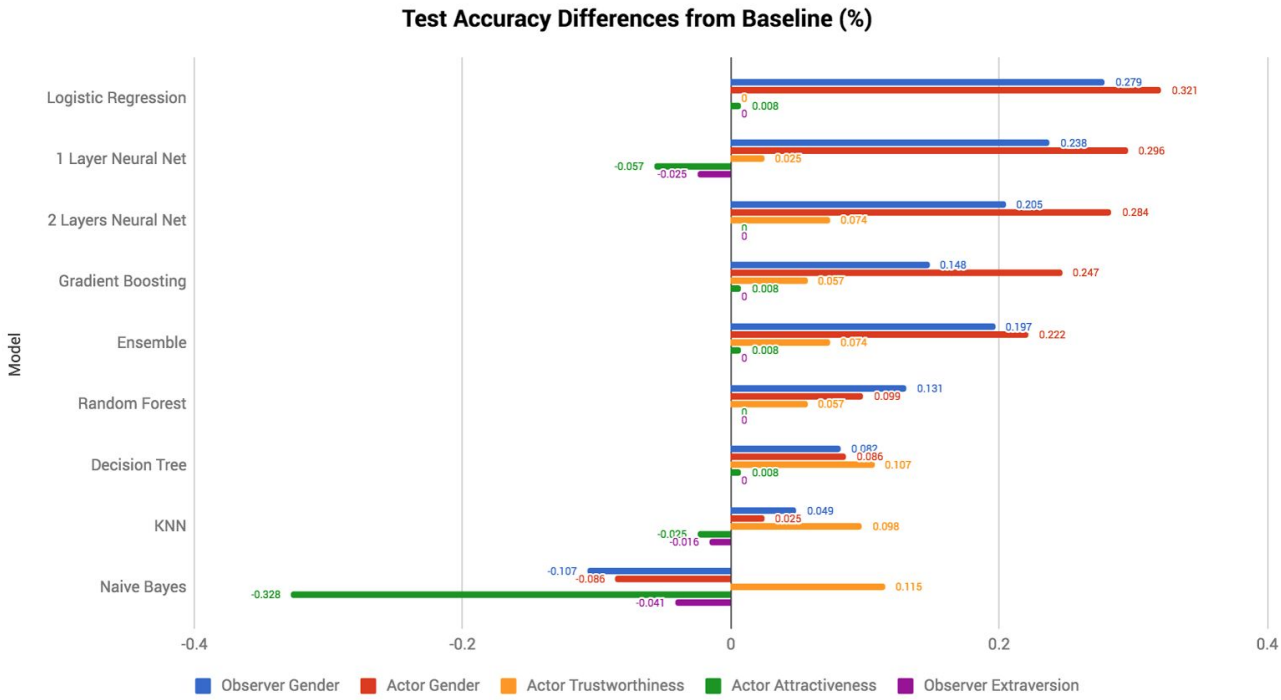
The final results are calculated on a held-out dataset, which was untouched during 10-fold cross-validation.

Below is the agreement matrix of all the different models predicting observer gender, the ensemble being a vote of all the models above. Expectedly, models with similar underlying algorithms appear highly correlated.

	Logistic Regression	Decision Tree	K-Nearest Neighbors	Naive Bayes	Random Forest	1 Layer Neural Net	2 Layers Neural Net	GradientBoosting	ensemble
Logistic Regression	1.00	0.63	0.70	0.81	0.70	0.63	0.72	0.62	0.82
Decision Tree	0.63	1.00	0.59	0.60	0.82	0.33	0.47	0.57	0.76
K-Nearest Neighbors	0.70	0.59	1.00	0.61	0.57	0.63	0.74	0.61	0.71
Naive Bayes	0.81	0.60	0.61	1.00	0.71	0.57	0.66	0.59	0.77
Random Forest	0.70	0.82	0.57	0.71	1.00	0.39	0.54	0.57	0.76
1 Layer Neural Net	0.63	0.33	0.63	0.57	0.39	1.00	0.77	0.73	0.55
2 Layers Neural Net	0.72	0.47	0.74	0.66	0.54	0.77	1.00	0.67	0.68
GradientBoosting	0.62	0.57	0.61	0.59	0.57	0.73	0.67	1.00	0.77
ensemble	0.82	0.76	0.71	0.77	0.76	0.55	0.68	0.77	1.00

Results and Discussion

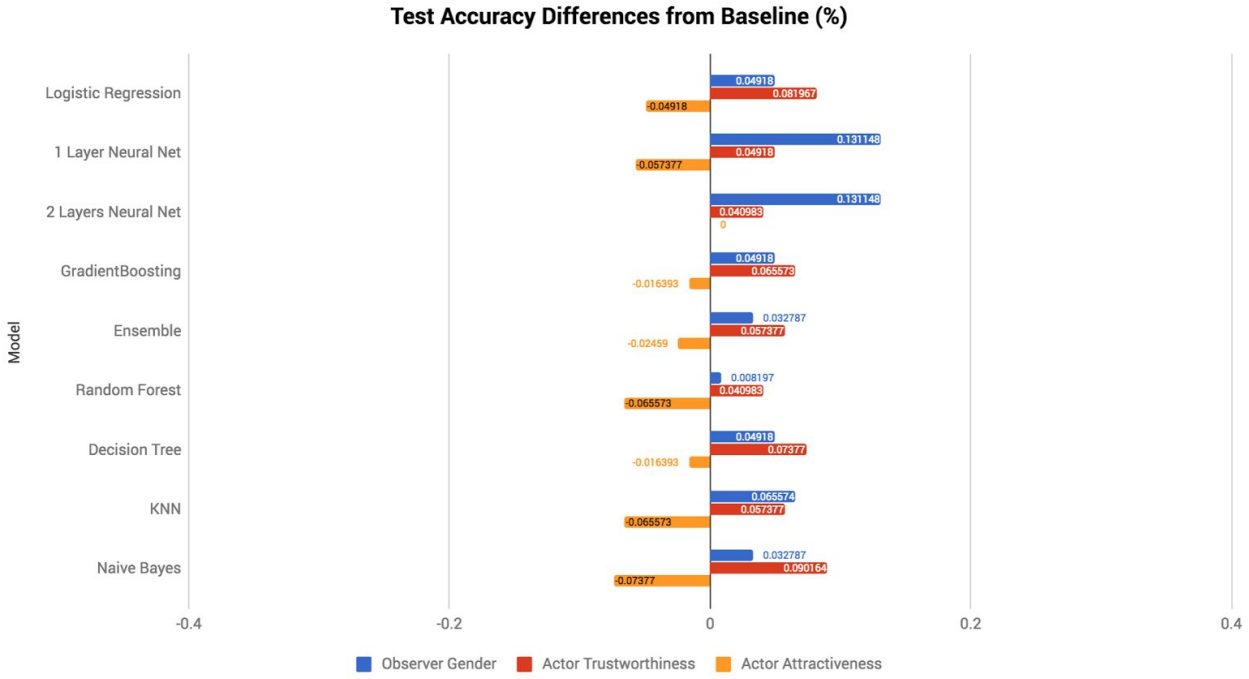
We used a variety of machine learning classifiers trained only on our eye tracking features to predict variables of interest and compared the results to a majority class prediction baseline. We converted personality scores and ratings into binary variables (<5 as low, >= as high for a given variable). In general our models perform well at predicting observer gender (28% over baseline), actor gender (32% over baseline), and trustworthiness ratings (11.5% over baseline). Logistic regression was the highest performing model for both observer and actor gender, while naive bayes performed best for trustworthiness ratings. We were unsuccessful at predicting threat, dominance, and attractiveness ratings of actors (shown as an example below) and all personality scores of observers, even extraversion (in purple) which we expected to be classifiable.



Classifier results trained only on eye tracking features.

The underlying assumption behind this analysis is that there is signal encoded in our unconscious eye movements that may expose information about ourselves, and our response to our perceptive environment. In some cases, this information may be private or sensitive.

Results from this analysis show that even with a relatively small dataset of eye tracking and pupillometric data, some target attributes and even subjective ratings can be predicted with reasonable accuracy. Notably, prediction of trustworthiness and attractiveness ratings via our eye tracking model outperformed a model trained on personality scores alone (see figure below).



Classifier results trained only on personality scores for observer gender, actor trustworthiness, and actor attractiveness.

We expect that prediction accuracy would increase when running similar analyses on a larger dataset of eye-tracked observations.

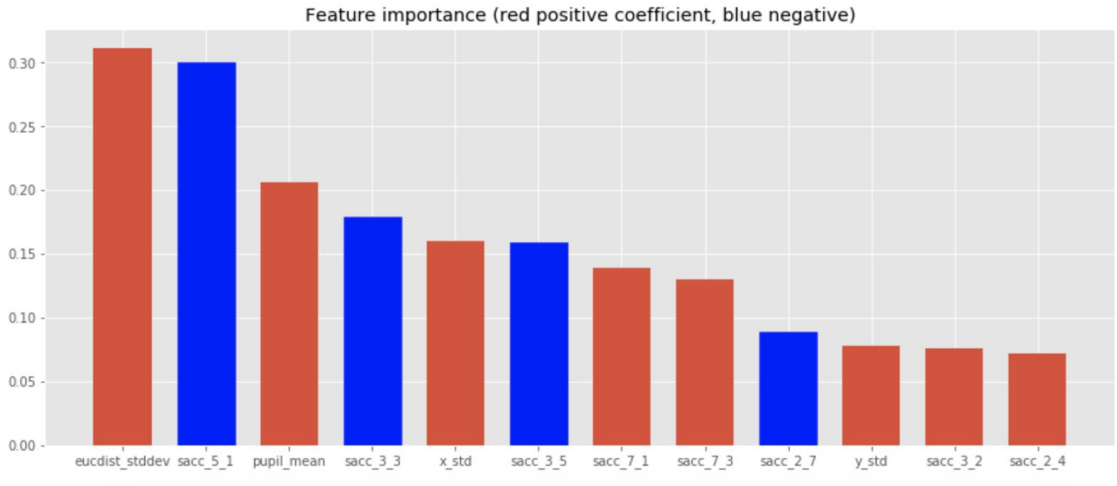
All results predicting actor-inherent attributes (e.g. face rating scores and gender) must be qualified by the fact that data from only 8 actors was available. Given the strong linkage between actor and face ratings, and the fact that it was necessary to train on data from all actors given the small sample size, these results are qualified by the fact that it is possible some of the resultant models learned to predict actor ID as a proxy for target labels. Because each of the 405 observations represent unique participants, models predicting personality ratings or subject gender is not subject to this concern.

Accuracy score for observer gender prediction

	Model	Held_Out_Score	Train_Score	Hyperparams
0	Logistic Regression	0.622951	0.734043	{'C': 1, 'penalty': 'l1'}
1	Decision Tree	0.598361	0.638298	{'max_depth': 1}
2	K-Nearest Neighbors	0.557377	0.709220	{'n_neighbors': 7}
3	Naive Bayes	0.549180	0.592199	{}
4	Random Forest	0.598361	0.737589	{'max_depth': 3, 'n_estimators': 7}
5	1 Layer Neural Net	0.573770	0.812057	{'hidden_layer_sizes': (2,)}
6	2 Layers Neural Net	0.581967	0.790780	{'hidden_layer_sizes': (4, 2)}
7	GradientBoosting	0.590164	0.751773	{'max_depth': 2, 'n_estimators': 9}
8	ensemble	0.655738	0.780142	Democracy without weights
9	baseline	0.500000	0.500000	NaN

Both train and test set are perfectly gender balanced, which justifies the 50% baseline.

The most important features in our best performant models (logistic regression and decision tree) were the standard deviation of the euclidean distance, followed by the probability of saccades moving from zone 5 (below chin) to zone 1 (left eye). This saccade coefficient is negative implying that men are more likely to saccade between these zones.



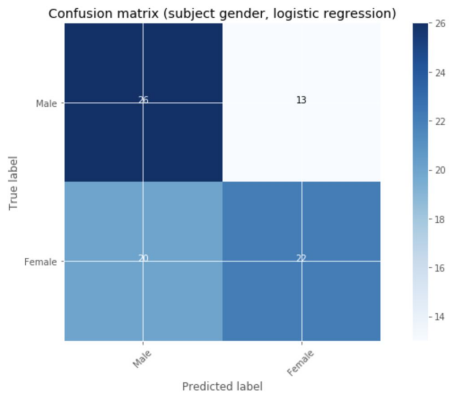
Feature importance (logistic regression predicting gender)

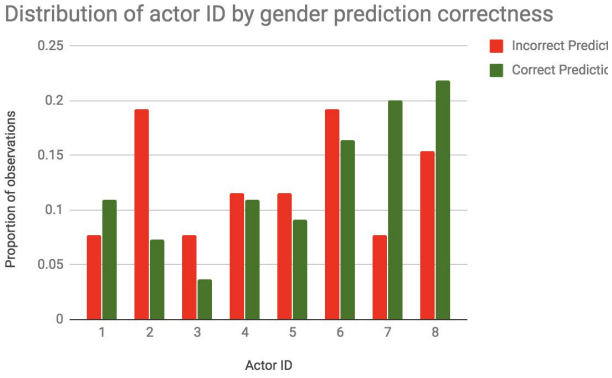
Interpreting the euclidean distance standard deviation isn't trivial, however we believe this feature encodes a general increase in the distribution of facial scanning behavior. Prior research has found that women scan a wider spatial distribution on a face (less focused on specific features), which is augmented by this finding as a high standard deviation of euclidean distance between readings could be representative of more or longer saccades (particularly because of the positive coefficient of this feature in our model).

The positive coefficient on the standard deviation on x-coordinate may also imply that one aspect of the increased distribution of women's scanning patterns is an increase in horizontal scanning behavior.

Error Analysis

The most performant individual model on predictions of observer gender was logistic regression with a full feature set of zone, saccade, and summary statistic features. As can be seen in the figure below, this model was slightly more likely to mis-predict male when the subject was female.





Confusion matrix of predictions of subject gender via logistic regression (left). Distribution of actor ID by gender prediction correct/incorrect. (right)

After segmenting the held out test portion of the dataset by gender prediction correctness, we see that some actors, notably actor 2, are over-represented in the mispredicted observations. We also see higher prediction success of observer's gender where the observation had a observer-actor gender match (71% for correct, 57% for incorrect), as well as observer agreeableness (51% highly agreeable for correct, 27% highly agreeable for incorrect).

Limitations

One distinct limitation was the small number of participants of only 405. One way to reconcile for the lack of participants could be to use each session (405 participants x 35 sessions) as an observation for further analysis. This would require a careful train/test split such that the same observer does not appear in both sets. However, another limitation was the fact that every actor was paired with an observer a different number of times. For example, actor ID #7 was paired with 76 different observers whereas actor ID #5 was paired with 30 different observers.

Actor ID	# Matched with Observer
7	76 (most paired with an observer)
5	30 (least paired with an observer)

Additionally, the dataset did not include information about the luminance of the room or the monitor during the gaze task. Pupil dilation has a strong relationship with exposure to light, and because we don't have data on how light was controlled in the space, there is a limit to the analyses we can conduct based on individuals' pupil diameter and changes during each session.

The amount of data available for each observation are also inconsistent due to the way the study randomized the length of each session (.3 to 10.3 seconds), there is a varied number of pupil sizes and eye coordinates per individual. As also mentioned in previous sections, we used the mean of these values per session and normalized them for better comparison between observers.

Conclusion

We have shown that machine learning techniques are capable of extracting predictive power from our target dataset of eye tracking and pupillometric data, though significantly limited by the sample size. The ability to leverage such a model to predict subject gender with reasonable accuracy leaves us optimistic that future studies will find significant linkages between eye tracking data and other attributes and subjective response metrics. Prior work has shown that (with specifically chosen stimuli) a wide variety of properties can be predicted from eye tracking data alone, including body mass index, ethnicity, a variety of personality traits, and sexuality. With the advent of consumer-grade virtual reality systems or mobile devices with built-in eye tracking modules, we expect large eye tracking datasets to become more prevalent. Such availability will be a boon to researchers, but also poses a significant privacy concern to consumers who may not be aware to what extent their usage of these emerging technologies may expose private information.

References

Behe, B. K., Fernandez, R. T., Huddleston, P. T., Minahan, S., Getter, K. L., Sage, L., & Jones, A. M. (2013). Practical field use of eye-tracking devices for consumer research in the retail environment. HortTechnology, 23(4), 517-524.

Binetti, Nicola, Charlotte Harrison, Antoine Coutrot, Alan Johnston, and Isabelle Mareschal. "Pupil dilation as an index of preferred mutual gaze duration." Royal Society open science 3, no. 7 (2016): 160086.

Blais, C., Jack, R. E., Scheepers, C., Fiset, D., & Caldara, R. (2008). Culture shapes how we look at faces. PloS one, 3(8), e3022.

Calderone, Julia, and Julia Calderone. (2015). “Eye Tracking in Google Glass: A Window into the Soul?” Scientific American. https://www.scientificamerican.com/article/eye-tracking-in-google-glass-a-window-into-the-soul/.

Coutrot, A., Binetti, N., Harrison, C., Mareschal, I., & Johnston, A. (2016). Face exploration dynamics differentiate men and women. Journal of vision, 16(14), 16-16.

Cognitive 3D. https://cognitive3d.com/.

Greene, M. R., Liu, T., & Wolfe, J. M. (2012). Reconsidering Yarbus: A failure to predict observers’ task from eye movement patterns. Vision research, 62, 1-8.

Hall, J. K., Hutton, S. B., & Morgan, M. J. (2010). Sex differences in scanning faces: does attention to the eyes explain female superiority in facial expression recognition?. Cognition & Emotion, 24(4), 629-637.

Kanan, C., Bseiso, D. N., Ray, N. A., Hsiao, J. H., & Cottrell, G. W. (2015). Humans have idiosyncratic and task-specific scanpaths for judging faces. Vision research, 108, 67-76.

Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., & Torralba, A. (2016). Eye tracking for everyone. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2176-2184).

Liebling, D. J., & Preibusch, S. (2014, September). Privacy considerations for a pervasive eye tracking world. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (pp. 1169-1177). ACM.

Mehoudar, E., Arizpe, J., Baker, C. I., & Yovel, G. (2014). Faces in the eye of the beholder: Unique and stable eye scanning patterns of individual observers. Journal of vision, 14(7), 6-6.

Perlman, S. B., Morris, J. P., Vander Wyk, B. C., Green, S. R., Doyle, J. L., & Pelphrey, K. A. (2009). Individual differences in personality predict how people look at faces. PloS one, 4(6), e5952.

Pupil Labs. https://pupil-labs.com/vr-ar/.

Singh, M. P., Poddar, M., Huang, R. Burgwin, N. (2016). Personality Characterization Using Eye Tracking. http://www.eecg.utoronto.ca/~jayar/CAM/eyamposter.pdf.

Tobii. https://www.tobii.com/.

Vassallo, S., Cooper, S. L., & Douglas, J. M. (2009). Visual scanning in the recognition of facial affect: Is there an observer sex difference?. Journal of Vision, 9(3), 11-11.

Appendix: Code for machine learning analyses

```
import pandas as pd
import numpy as np
import pickle

# Plot
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
matplotlib.style.use('ggplot')

# General Machine Learning
from sklearn.model_selection import train_test_split, KFold, cross_val_score,
cross_validate, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
# Binary Models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, VotingClassifier,
GradientBoostingClassifier
from  sklearn.neural_network import MLPClassifier
# Real Value Model
from sklearn.linear_model import LinearRegression
```

```
random_state = 5
held_out_size = 0.3
n_splits = 10

df = pd.DataFrame.from_csv('nobel_prize_data_2017.csv', sep='\t')
df.columns

features = ['zone_1', 'zone_2', 'zone_3', 'zone_4', 'zone_5', 'zone_6',
'zone_7', 'sacc_1_2', 'sacc_1_3', 'sacc_1_4', 'sacc_1_5', 'sacc_1_6',
'sacc_1_7', 'sacc_2_1', 'sacc_2_3', 'sacc_2_4', 'sacc_2_5', 'sacc_2_6',
'sacc_2_7', 'sacc_3_1', 'sacc_3_2', 'sacc_3_4', 'sacc_3_5', 'sacc_3_6',
'sacc_3_7', 'sacc_4_1', 'sacc_4_2', 'sacc_4_3', 'sacc_4_5', 'sacc_4_6',
'sacc_4_7', 'sacc_5_1', 'sacc_5_2', 'sacc_5_3', 'sacc_5_4', 'sacc_5_6',
'sacc_5_7', 'sacc_6_1','sacc_6_2', 'sacc_6_3', 'sacc_6_4', 'sacc_6_5',
'sacc_6_7', 'sacc_7_1', 'sacc_7_2', 'sacc_7_3', 'sacc_7_4', 'sacc_7_5',
'sacc_7_6', 'sacc_1_1', 'sacc_2_2', 'sacc_3_3', 'sacc_4_4', 'sacc_5_5',
'sacc_6_6', 'sacc_7_7', 'eucdist_stddev', 'pupil_mean', 'xy_std_rat', 'x_std',
'y_std', 'X_mean', 'Y_mean']

target = ['gender']


scoring = ['accuracy']

X = StandardScaler().fit_transform(df[features])

X_train, X_held_out, y_train, y_held_out = train_test_split(X,
                                                            df[target],
                                                            test_size =
held_out_size,
                                                            random_state =
random_state)
y_train = np.ravel(y_train)

kfold = KFold(n_splits = n_splits,
              random_state = random_state)

values, counts = np.unique(y_train, return_counts = True)
majorityClass = values[np.argmax(counts)]
baseline = accuracy_score(y_train, np.repeat(majorityClass, len(y_train)))
print(np.unique(y_train, return_counts = True))
np.unique(y_held_out, return_counts = True)

binary_models = [LogisticRegression(),
                 DecisionTreeClassifier(),
                 KNeighborsClassifier(),
```

```
                GaussianNB(),
                RandomForestClassifier(random_state = random_state),
                MLPClassifier(learning_rate='adaptive', learning_rate_init =
0.1, random_state = random_state),
                MLPClassifier(learning_rate='adaptive', learning_rate_init =
0.1, random_state = random_state),
                GradientBoostingClassifier(min_samples_leaf=3, random_state =
random_state)]

parameterScope = [{'penalty' : ['l1', 'l2'], 'C' : [0.001, 0.005, 0.01, 0.05,
0.1, 0.5, 1, 2, 5, 10]},
                 {'max_depth' : list(range(1, 5))},
                 {'n_neighbors' : list(range(3, 25, 2))},
                 {},
                 {'max_depth' : list(range(1, 5)), 'n_estimators' : range(3,
10, 2)},
                 {'hidden_layer_sizes' : [(2,), (3,), (4,)]},
                 {'hidden_layer_sizes' : [(3,2,), (2,2,), (4,2),
                                          (2,3,), (2,4,)]},
                 {'max_depth' : list(range(1, 4)), 'n_estimators' : range(3,
10, 2)}]

binary_names = ['Logistic Regression',
                'Decision Tree',
                'K-Nearest Neighbors',
                'Naive Bayes',
                'Random Forest',
                '1 Layer Neural Net',
                '2 Layers Neural Net',
                'GradientBoosting']

fitted_binary_models = []
performance = pd.Series()
bestParam = pd.Series()


for model, parameters, name in zip(binary_models, parameterScope,
binary_names):
    parameterEstimator = GridSearchCV(model, parameters, cv = kfold, scoring =
scoring[0], refit = True, return_train_score=False)
    parameterEstimator.fit(X_train, y_train)
    performance[name] = parameterEstimator.best_score_
    bestParam[name] = parameterEstimator.best_params_
    fitted_binary_models.append(parameterEstimator.best_estimator_)

bestParam

# Set weight depending on performance
```

```
weights = pd.Series(index = performance.index, data = 1)
'''
performance.sort_values(inplace = True)
position = 1
for modelName in performance.index:
    weights[modelName] = position
    position = position + 1
'''
weights['Logistic Regression'] = 3

weights = None
ensemble_models = list(zip(binary_names, fitted_binary_models))
ensemble = VotingClassifier(ensemble_models, voting = 'hard', weights =
weights)
bestParam['ensemble'] = 'Democracy without weights'

models = fitted_binary_models + [ensemble]
models_names = binary_names + ['ensemble']
# for matrix agreement
predictions = pd.DataFrame()

model_results = pd.DataFrame()
for name, model in zip(models_names, models):
#    scores = cross_validate(model, X_train, y_train, cv=kfold, scoring =
scoring, return_train_score = True)
    model.fit(X_train, y_train)
    result = {'Model' : name,
              'Train_Score' : model.score(X_train, y_train),
              'Held_Out_Score' : model.score(X_held_out, y_held_out),
              'Hyperparams' : bestParam[name]}
    model_results = model_results.append(result, ignore_index = True)
    # model prediction only used for agreement matrix
    predictions[name] = model.predict(X_train)

result = {'Model' : 'baseline',
          'Train_Score' : baseline,
          'Held_Out_Score' : accuracy_score(y_held_out,
np.repeat(majorityClass, len(y_held_out)))}
model_results = model_results.append(result, ignore_index = True)

columns_order = ['Model', 'Held_Out_Score',  'Train_Score', 'Hyperparams']#
'CV_TrainAccuracy', 'CV_TestAccuracy'
model_results[columns_order]
```