**Method**

**Participants**

The current sample consists 169 participants. 131 participants were allocated to the training set and the other 38 participants were allocated to the test set. [**later** we will have to relate to the characteristics of each set, that is, we allocated participants in a way that keeps the distribution of LSAS scores in each set similar].

~~Half of the participants were high socially anxious and the other half low socially anxious.~~ ~~This 1:1 ratio was maintained in the training and test sets.~~

* ~~Using stratified splitting, we kept aside 30 participants of the data set as test set.~~

**Measures**

**Leibowitz Social Anxiety Scale (LSAS).** The LSAS is a psychological inventory comprised of 24 items depicting social situations that might cause fear in socially anxious individuals (Liebowitz, 1987). Participants rate the fear and avoidance of each social situation on a scale ranging 0-3. The LSAS has good psychometric properties (G. Heimberg et al., 1999).

**Eye-tracking data.** We collected eye tracking data from a free viewing task as the one reported by Lazarov et al., 2016. This task has shown differential gaze patterns between high and low socially anxious individuals [REF]. Briefly, after a short calibration procedure, participants were instructed to freely watch face matrices that included both neutral and threat-related faces for about three minutes. Each trial began with the appearance of a fixation cross. Once participants fixated their gaze at the cross, a face matrix appeared instead for six seconds. Each matrix was comprised of 16 cells (4X4), each containing a face. Half of the faces were neutral, and half were disgusted. In addition, half of the cells presented women and half men. The face stimuli were adapted from Karolinska Directed Emotional Faces database (KDEF; Lundqvist et al. (1998)). Each participant viewed 30 matrices in total from distance of 70cm, the sampling rate was 500Hz. The output data included fixation duration, fixation location (i.e., on which specific stimulus the fixation was, namely, threat or neutral area of interest), and pupil size during fixation.

**The classifier**

Using machine learning model creation, we aimed at developing an algorithm that could classify high and low socially anxious individuals (see Figure X). To this end, we collected eye tracking data from 130 participants with various LSAS scores (Table x - later). Each data point of the eye tracking output represents a single fixation and is composed of fixation duration, fixation coordinates and pupil size. In addition, each participant was labeled as high or low on social anxiety according to the LSAS score, with 50 as the cutoff score [REF – in Lazarov’s article there is justification for this cutoff as good in discrimination of high and low].

**Predictors.** We created 25 features from the eye-tracking raw data (fixation duration, fixation coordinates, and pupil size) using them as predictors for the model (see Table XX in supplementary materials). The predictors were calculated using aggregation functions such as standard deviation, mean, or total sum (divided by total number of matrices) over specific areas of interest (i.e., threat or neutral faces). None of the features had dependency on the number of matrices (i.e., the features calculation could be generalized for any amount of matrices). We removed from the data the first fixation of each matrix and every fixation with duration lesser than 100ms. And removed matrices with less then 4 fixation in them. There were no missing data in the eye-tracker’s.

[explain here on the model building first]

We are choosing our model from an various of machine learning algorithms, and choosing the best fitting one. And using the same process for choosing the model's hyperparameter.

**Estimating generalization.** We divided the 100 samples of the training set in a pseudo-random stratified method, creating holdout set of 10 samples. The training set contained 90 sample, equally divided to the low and high social anxiety groups.We used grid search over the training set, and picked the set of hyperparameters with the highest cross validation accuracy score.Due to the relatively small training set, we used Leave One Out cross validation method. To achieve the most accurate estimation of the generalization, we repeated the process 100 times and used the average accuracy score as our cross-validation score. To avoid overfitting, after estimating the accuracy of the training set, we trained the chosen model on the full training set and tested the model’s accuracy on the holdout set. As a final step, we calculated the model’s accuracy on the test set.

* algorithm selection; justify why we chose the specific algorithms we used and not others

Our pipeline is composed of the following steps: feature scaling (if needed), feature selection, and classifier.

We used feature selection to reduce the dimensionality of the data, it is useful technique especially in a relatively small training set.   
our feature selection had two sequential steps, the first is correlation threshold transformer – it removes randomly one of each two features that their correlation is greater than the threshold parameter.

The second step is Recursive Feature Elimination, an algorithm used to select optimal subset of features, using a model that enable scoring the feature importance.

The classifier is an algorithm capable of building a model representing the target as a function of the data.

We tested several classifier families:

* random forest
* gradient boosting (using xgboost library)
* support vector machine – good for reletivly small data set

**Choice of classifier families/ The final classifier**

* We constructed Pipeline with Sklearn python library, using custom transformer if needed. The pipeline is:

1. Correlation threshold.
2. Feature selection - Recursive feature elimination. With random forest as estimator. n\_estimaors=100.
3. Classifier – XGBoost

* No feature scaling or handling missing values was needed.
* Rewrite
* hyperparameters tuning
* collapse with section above?
* test set.
* Not sure

General notes: order as in the presentations so far, justify why we chose the specific algorithms we used and not others

A general explanation about the pipeline…