We considered three major classes of models in our research: Logistic Regression, SVM and Random Forests. In all three we used the binary classification version of the algorithm.

We started with Logistic Regression for a few reasons. (1) The algorithm is fast, eg compared to SVM. (2) Even after feature selection, one-hot encoding resulted in many features. This initially made us wary of decision trees which if done incorrectly can overfit the training data. (3) In our exploratory phase we found a number of features with high conditional probabilities of “y given x” and this is what logistic regression models. (4) At the onset we wanted to establish a baseline classifier and logistic regression is simple in that the model assumes a single linear decision boundary.   
  
We also experimented with SVM, using both linear and Gaussian (RBF) kernels. We focused our effort on the RBF kernel because (1) our available training set had more than 100,000 observations and we only had around 500 features after one-hot encoding and (2) . The results were satisfactory with an error rate slightly below our final classifier but the iteration cycle was too long. Some actions we could have taken to shrink the iteration cycle are (1) focus on linear kernel and (2) drastically shrink the training set.

We then turned to ensemble methods. We quickly tried AdaBoost with Decision Trees of depth 1 as the weak learner but the results were only slightly better than Logistic Regression and definitely worse than SVM. We then turned to Random Forests which are ensembles of Decision Trees. What drew us to Random Forests is how it uses randomness. Unlike Decision Trees which use all features to make a split decision, Random Forests only consider a random subset (of size k, eg k=sqrt(number features)) of features at each split decision up to . This gave us the power to test different subsets of features, albeit implicitly. From the start Random Forest performed on par with SVM, but unlike SVM the runtime is fast and provided us with short iteration cycles to tune a few parameters.