CIS 520 Final Project Report

Gabriel Solomon and Trevor Wexner

December 10, 2018

Introduction

The state of affairs of healthcare in the United States is, unfortunately, quite dire. Despite leading the developed world in healthcare spending per capita, at $10,224 per person, disease burden is higher in the U.S than comparable countries – which means the U.S is spending more money for worse health outcomes than any other developed nation [1]. This gap serves as a major indicator for the inefficiencies present in our healthcare system. While healthcare providers, hospitals, and doctors are all working hard to provide quality care for patients, there remains room to improve; a priori identification of patients and regions that will require specific health interventions in the future would dramatically improve the system.

Given this disconnect, this project focused on predicting nine key health behaviors and outcomes at the county level. These outcomes included obesity percentage, mental health levels, and substance abuse metrics, among others. The data used to make these predictions came from a combination of publicly available demographic data and socioeconomic factors as well as tweets mapped to 2,000 LDA topics. There were 21 socioeconomic and demographic predictors along with the county-level tweet frequencies for each LDA topic, giving each of the 1,019 counties in the dataset 2,021 total predictors. Our team worked to create a model that predicted healthcare outcomes, with a scaled error metric serving as the measure of prediction strength. These modeling approaches could help healthcare practitioners target interventions in a more systematic, county-level manner, improving the quality of healthcare nationwide.

Leaderboard Submission

Our leaderboard submission was identically the elastic net model we used to satisfy the discriminative modeling approach. Please see below for a full explanation of this method; put simply, we took logs of the tweet frequency data and ran nine hyperparameter-tuned elastic net linear regressions to generate an outcome prediction matrix. The training error for this approach was .0627, and the testing error was .0775 - see the "Analysis of Methods" section for more information on performance.

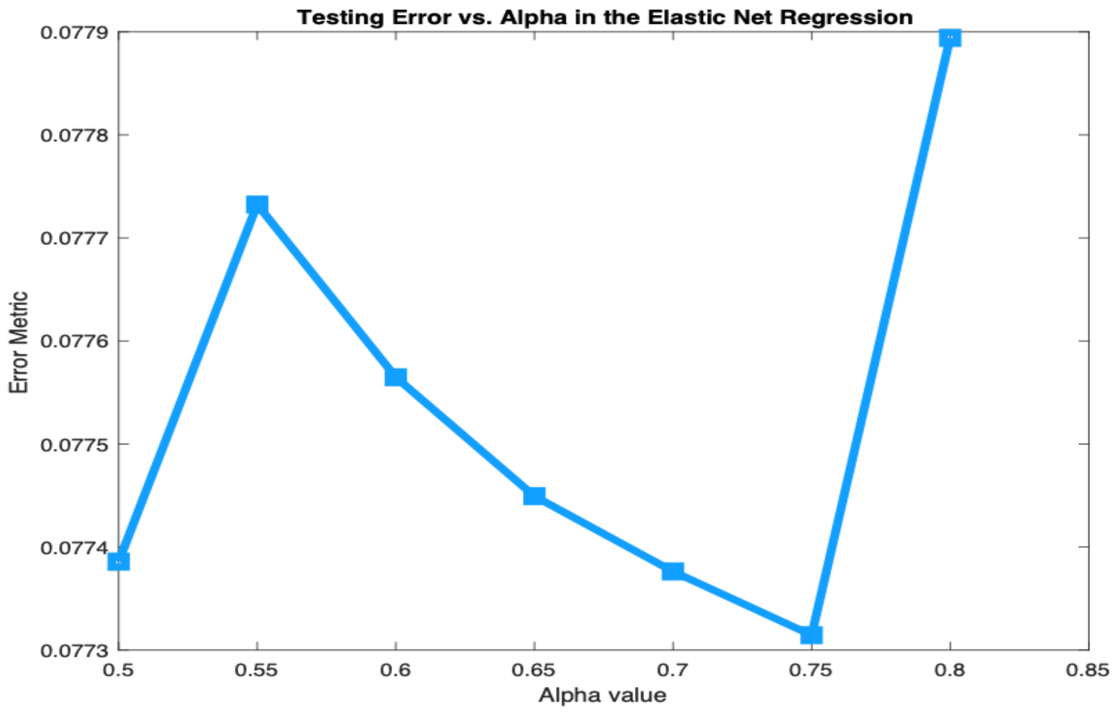
Modeling Approaches

In order to find the best possible model and explore the wide range of topics covered in class, our team created models using four modeling approaches: discriminative, instance-based, generative, and innovative. In this section, we will detail the logic behind the models created for each category as well as the steps taken to construct these models. The final training and testing error for each method will be reported in the "Analysis of Methods" section.

Discriminative Method

For the discriminative method, we chose to use an elastic net regularized linear regression. We opted for this approach because of the large number of predictors present in the input data (2,021) relative to the number of observations (1,019). Given nine separate outcome variables, we ran nine separate elastic net regularizations and generated nine sets of coefficients, one for each outcome. The limitation of this approach is that it does not account for correlation across the outcome variables, an issue we attempted to remedy with our "novel" approach. In order to prepare the data for an elastic net regression, we first took the log_{10} of every tweet category probability under the LDA. Since we were constructing a linear regression, it did not make sense to consider the probabilities as they were given, since an increase of topic frequency from .001 to .002 is only an addition of .001 in linear space, but this represents a doubling of the probability. Thus, we worked with these topic frequencies in log space, making them more amenable to linear regression. After taking the log of the tweet probabilities, this data was added back to the unchanged socioeconomic and demographic factors to create adjusted observation vectors suitable for elastic net regularization.

Given that elastic net finds $\underset{\beta_0,\beta}{\operatorname{argmin}} \frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}(\frac{(1-\alpha)}{2}\beta_j^2 + \alpha\,|\beta_j|)$, α is a key hyperparamater to tune in this model, as it indicates the balance between ridge and lasso penalties. Given our desire for feature shrinkage, we chose to test α (where the α value for each of the nine elastic net regressions was the same) between .5 and .8, over the values [.5, .55, .6, .65, .7, .75, .8], as this ensured we were taking advantage of LASSO's propensity for zeroing out coefficients and truly shrinking the number of predictors. After dividing the given data into 900 training observations and 109 testing observations, the testing error plotted for each α was as follows:



Since $\alpha = .75$ produced the lowest error on a holdout test set, we chose to use that value for each of the nine elastic net regressions.

Another key hyperparameter in this model was the λ value, or the size of the regularization penalty. The authors of *The Elements of Statistical Learning* claim "We often use the 'one-standard-error' rule when selecting the best model; this acknowledges the fact that the risk curves are estimated with error, so it errs on the side of parsimony" [2].

It was especially important to choose a parsimonious model with so many predictors in the input data. For this reason, each elastic net regression used the coefficients for the value of λ that lowered cross-validation error minus one standard error, nicknamed $\lambda 1se$. We chose this value instead of the λmin value, which would provide the best CV error but may be overfitted.

After selecting these hyperparameters, we computed the elastic net coefficients using the "lasso" function in Matlab and used them to generate linear regression for each of the nine outcome variables, combining them into one outcome vector for the sake of computing testing and training error, which is reported below.

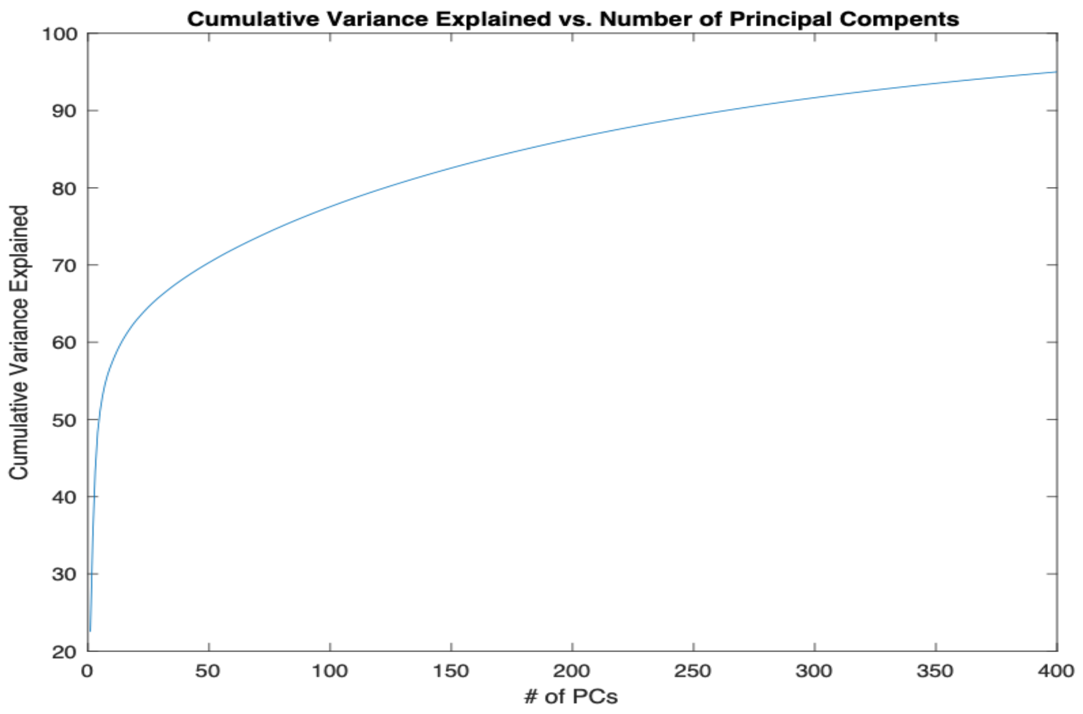
Generative Method

We continued to focus on feature selection and shrinkage in our generative method, where we used PCA on the data (using the "pca" function in Matlab) and then generated a random forest (using the "Random Forest" function) on the PCA-transformed data. In conducting PCA, we chose between three approaches: conducting PCA on the entire dataset, performing PCA on just the tweet topic data, and implementing PCA on tweet data and inputting the PCA-transformed tweet data into a PCA with the demographic information.

In all three approaches, the PCA data was input into nine random forests, one for each outcome variable, with 500 decision trees in each forest and $\sqrt{p}$ number of predictors randomly considered by the random forest at each split. These hyperparameters are commonly accepted and therefore were not tuned.

Approach 1: PCA on All Predictors

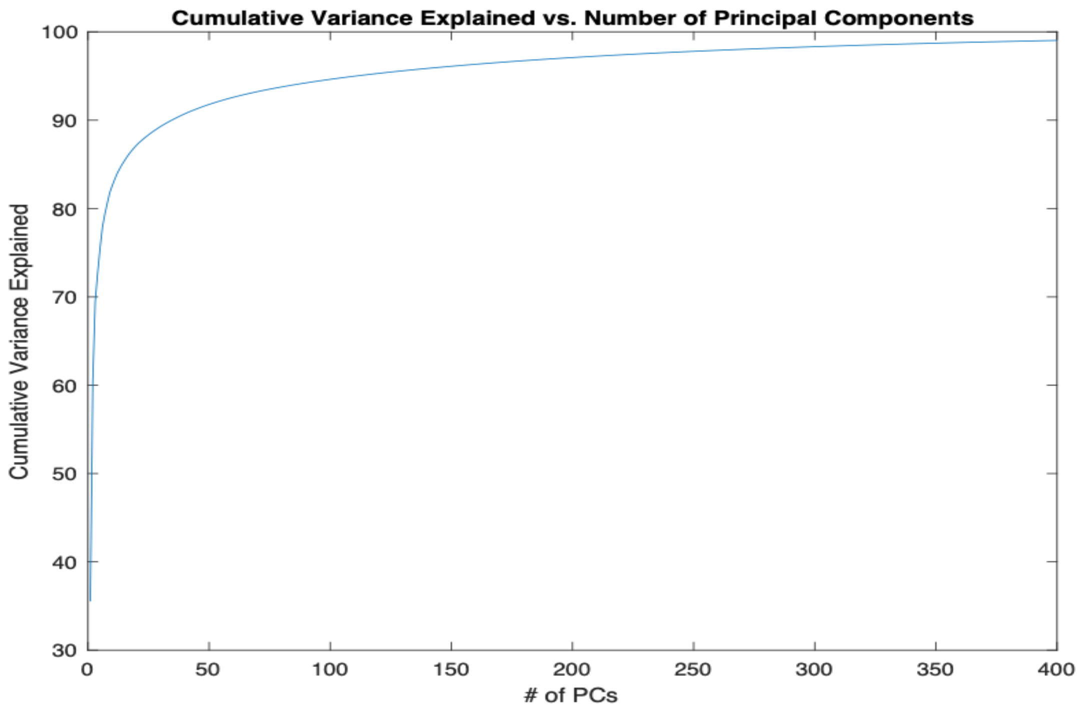
Running a PCA on all the predictors, including both socioeconomic variables and tweet-level data, was done under the assumption that principal components could help to explain variance in both the demographic information and tweet data simultaneously. The key question in this step was the number of principal components to include in the random forest; graphing over the possible number of PCs is shown here:



In an effort to shrink the number of predictors, we chose to use the first 123 principal components, which account for 80% of the variance in the data, in the random forest. Though it is standard to include enough principal components to account for 90% of the variation in the data, this would have meant including over 300 principal components, which we deemed to be too large. Thus, we kept 123 principal components for this approach.

Approach 2: PCA on Tweet Data Only

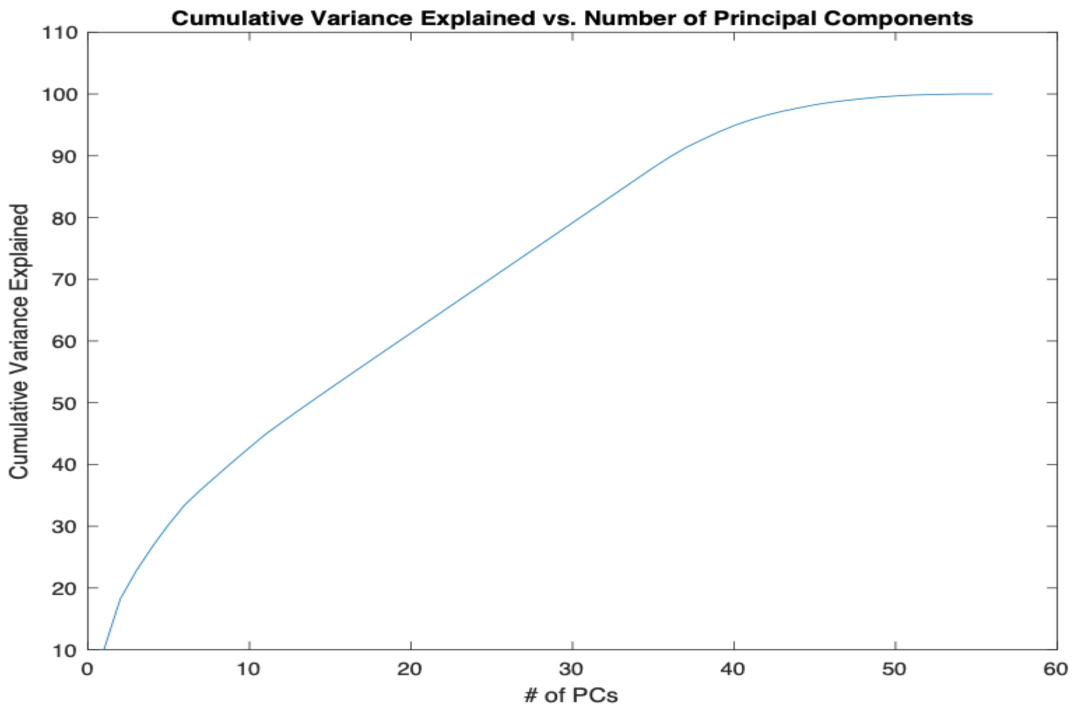
Running a PCA only on tweet data (columns 22-2021), resulted in a cumulative variance explained graph that looked as follows:



Since the first 35 principal components were able to explain 90% of the tweet-level variance in the data, we combined the unchanged socioeconomic data with the top 35 principal component scores as the input into a series of random forests, as described above. This approach seemed the most logical, as we expect counties with similar lifestyles, environments, and upbringings to tweet about similar topics â with each county having underlying factors that explain tweet frequencies.

Approach 3: PCA on Demographics and PCA-Transformed Tweet Data

Approach 3 was an amalgamation of Approach 1 and 2; after combining the PCA-transformed tweet data with the raw demographic data, we transformed that combined data through PCA. This reasoning behind this approach was that the tweet PCA data had far lower dimensionality than it had originally, making the drivers of variance across the combined demographics and tweet data easier to find. This was reflected in the variance explained chart:



Again, 35 Principal Components were enough to account for 90% of the variance, so the top 35 PC scores were used in the random forest models.

Approach Comparison

After transforming the input data through each of the three approaches described above, we modeled nine separate random forests and created a combined prediction vector on the transformed data in order to pick the best approach. Data was split into 900 training samples and 109 test samples, and the errors were calculated, as shown in the following table:

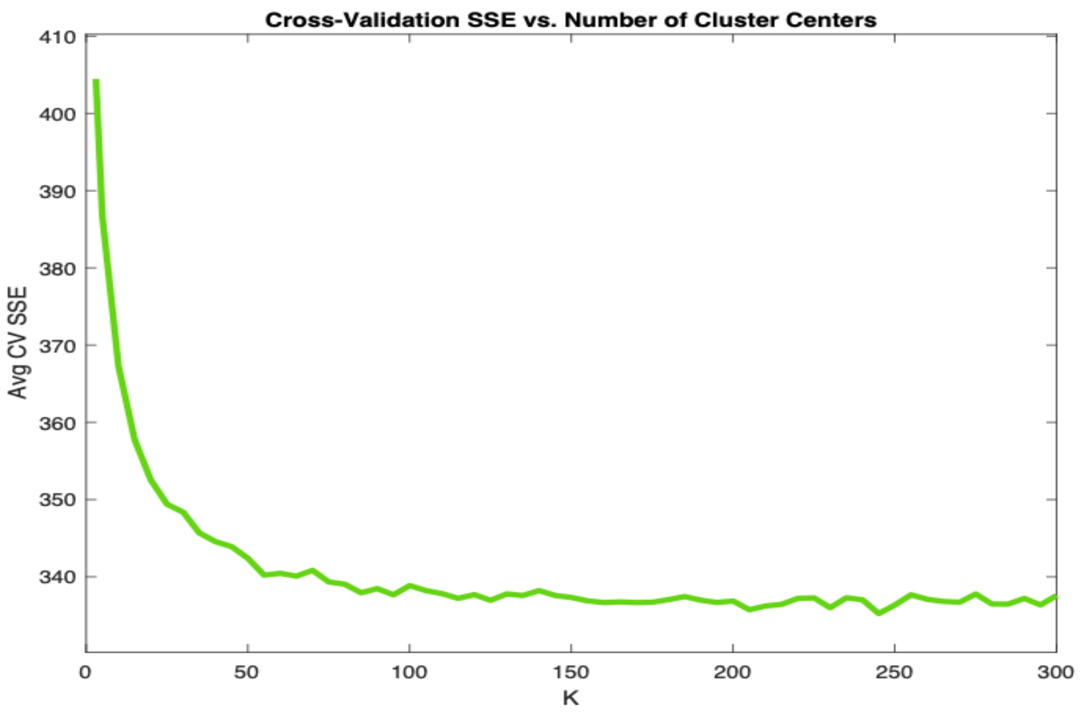
Approach	# PCs	Random Forest Testing Error
PCA on All Data	123	0.1061
PCA on Tweet Data Only	35	0.0872
PCA on Tweet PCA + Demographics	35	0.0943

After testing each approach, it was clear implementing a PCA on the tweet data alone was the best approach in terms of minimizing the out of sample error metric. Thus, the PCA and Random Forest model under Approach 2 was the chosen final model for the Generative Method section, and will be compared to the other models in the Analysis section.

Instance-Based Method

For the instance-based method, we built and ran a Radial Basis Function model that uses K-means clustering as the basis for selecting K cluster centers around which we computed a Gaussian Kernel regression. We chose this approach because RBFs are a regression technique that allows us to do dimensionality reduction simultaneously with prediction, which is important in this case, as the number of predictors was double the number of observations. Secondly, we were operating under the assumption that there was probably some set of latent classes for "population types" for different counties. We wanted to use clustering to capture these different population class types, and engage in intelligent prediction of the outcomes based on each cluster. Rather than just taking a simple average, using a radial basis function allowed us to build in a more complex relationship between clusters and predictions. K-means allowed for quick, understandable data separation and was therefore used to separate the data into clusters.

In order to find the best K number of centers for the K-means clusters, we first standardized the input data and then tested 61 different K values, ranging from 3 to 300. Each was tested using 10-fold cross validation, using the loss function of the average of the L2-norm distance from the out-of-sample points to the nearest cluster center. While this loss function is guaranteed to be minimized by the maximal number of clusters tested (the more clusters there are, the closer some cluster will be to an out-of-sample point), we instead focused on finding the "elbow" of the graph, i.e. the number of clusters where improvement began to dramatically slow. These calculations were made using the "kmeans" function in Matlab. The graph of average distance to the closest cluster center against the number of clusters is as follows:

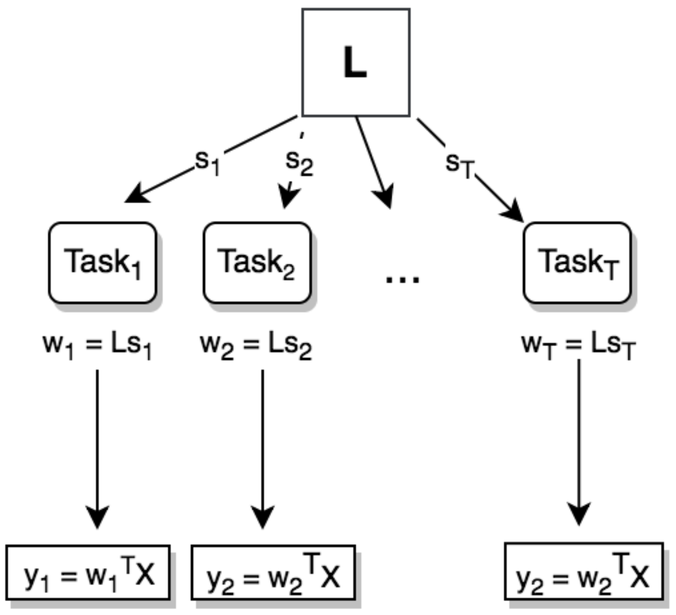


This "elbow" was achieved at K = 80 clusters, so we used 80 cluster centers to generate a K-means Gaussian kernel. This Gaussian structure represents an assumption of normality of data distribution which is certainly not fully reflected by reality, but may accurately represent the data for these purposes. AS NOTED IN THE LITERATURE, it is standard to use the average distance between nearest neighbors as the kernel width in a RBF kernel. This kernel width served relatively well for our purposes, so we created a Matlab function that calculated the average nearest-neighbor distance and used this to calculate kernel width.

Once the K means and kernel width were selected, we generated weights for a RBF regression using the standard linear regression formula $\beta = (Z^T * Z)^{-1} * (Z^T * Y)$, where Z is the kernel matrix from each point in X to each of the K cluster centers. We then had our model $Y = \beta^T Z$.For prediction on a test set, we took in a new set of observations, X_{test}, and computed the kernel matrix, Z_{test}, from the test set to the trained cluster centers. We then predicted $\hat{Y} = \beta^T Z_{test}$. Using this RBF regression, we hoped to capture the complex relationship between clusters and outputs. This technique will be analyzed below.

Novel Method

In order add an additional level of creativity to the models and take advantage of the fact that the Y variables are correlated, we tried to build a multitask learning approach that allowed each model of a different Y to share information. Our approach was inspired by Professor Eric Eaton's algorithm called ELLA [3]. In Professor Eaton's approach, a set of T tasks can be expressed in terms of a latent knowledge basis of dimension $< T$. Specifically, the weight vector for any task, $\mathbf{w_t}$, can be expressed in terms of a latent basis, $\mathbf{L}$, multiplied with some projection vector, $\mathbf{s_t}$. That is, for all tasks $t \in \{1, \ldots, T\}$, we have $\mathbf{w_t} = \mathbf{Ls_t}$. The idea behind using a basis $\mathbf{L}$ of dimension $\dim(\mathbf{L}) = k < T$ is that the latent basis expresses the true overlapping set of features that influences the outcome of each task. This structure of using a latent basis is described the figure below:



In order to solve for $\mathbf{L}$ and $\mathbf{s_t}$, we first solved for the optimal regression weights of each task independently (using the elastic net technique described in the "Discriminative Method" section). These separate regressions yielded 9 separate vectors of coefficients. We then put these 9 vectors into a matrix and used PCA to find a shared latent structure among the coefficients of each of the regressions. We took $\mathbf{L}$ to be the first k principal components of this combined weight matrix and $\mathbf{s_t}$ as the corresponding scores. We then predicted each task using this shared basis $\hat{Y}_t = \mathbf{Ls_t} * X$. After examining explained variance of different principal components, we chose $k = 2$ of our dimension of our latent basis. ($k = 2$ principal components explained over 98% of the variance in the coefficient matrix). Note that we did not do any additional hyperparameter tuning for this method because we used our already tuned elastic net models as inputs.

We derived this method of indirectly obtaining $\mathbf{L}$ and $\mathbf{s_t}$ from prior calculations, rather than directly solving for them in a newly posed problem due to computational constraints. That is, we did not use the ELLA code directly, nor did we implement a similar gradient descent algorithm that simultaneously solved a joint optimization problem across all dimensions to find the best $\mathbf{L}$ and $\mathbf{s_t}$ due to time limits on the submission.

Analysis of Methods

After creating generative, discriminative, instance-based, and novel methods, we compared their performance in order to understand the key drivers of health outcomes at the county level, as well to see the strongest modeling techniques. The following table details the training and test error for the models described above, with each model being trained on 900 given observations and tested on a holdout set of 109 observations:

Model Type	Training Error	Testing Error
Discriminative: Elastic Net Regression	.0627	.0775
Generative: PCA + Random Forest	.0930	.0980
Instance-Based: K-Means + RBF	.1316	.1639
Novel: Multi-Task Learning Approach	.8764	.8844

The discriminative elastic net method was clearly superior to the other methods tested, with testing error 21% lower than any other method. We conjectured that this improved performance stemmed from a few factors, namely: lower model complexity, our team's familiarity with the model, and the structural form of the elastic net. The multi-task learning approach certainly has merit, but our lack of experience and relatively naive approach of using PCA to create multi-task latency led to extremely inaccurate results. A proper implementation of the multitask learning problem that simultaneously solved for the optimal latent basis across all tasks probably would have yielded better results. Additionally, while the K-Means algorithm is a solid way to cluster observations, the complexity of this approach combined with our lack of human intuition as to cluster creation made this model produce somewhat lackluster results.

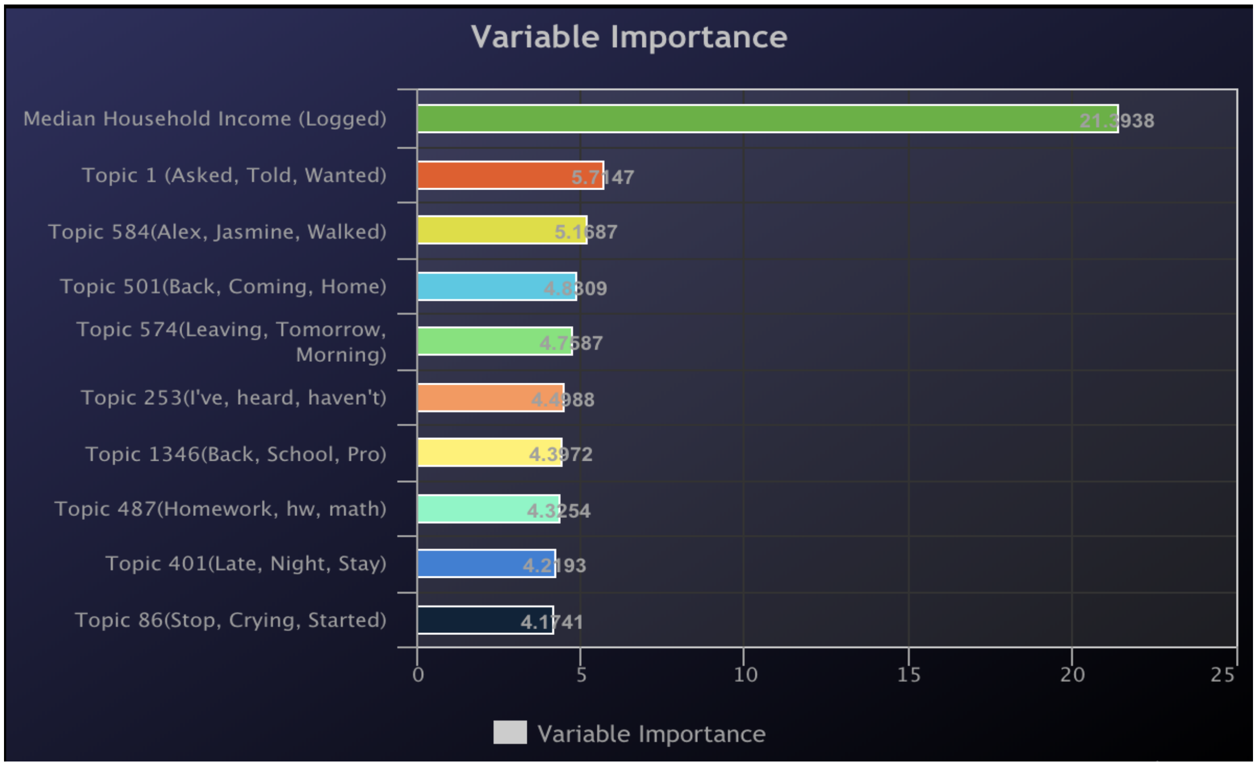
The Elastic Net was simple and effective, especially given the log-transformed data, and the linear form approximated the tasks well intuitively - health outcomes can easily be thought of as a weighted combination of the different inputs into the model. For this reason, the "simplest" model actually did the best; this is both a reassuring development (we can understand the world!) and a slightly disappointing result (we could have solved this problem after three weeks of class). The level of optimism concerning our results is left to the reader.

Model Interpretation

First of all, by looking at both our models and the impressive models created by our classmates, it is clear that regional factors, such as demographic information and socioeconomic factors, along with tweet topic data allowed us to create models that were quite accurate at predicting important health outcomes. This indicates that the predictive analysis missing from the U.S healthcare system is not a far-off pipe dream - these tools exist and are at our disposal.

Just knowing that the models perform well in general, however, stops short of our ultimate goal, which was understand where time and effort is best spent in order to improve health outcomes. To do this, we must drill down into what the models indicate are most important predictors of public health outcomes, match them against our intuitive understanding of their importance, and provide clear, useful takeaways for public health officials.

First, we looked for the largest coefficients by absolute value in our leaderboard submission, the elastic net model, as a proxy for variable importance. Since this model performed the best, it was logical to begin our search for variable importance at this stage. We averaged the coefficients for the input variables across the nine elastic net regressions and computed the largest absolute values. These variables are plotted below, with the variable name and top three tweets included in parentheses (if the variable was topic frequency):



While one should strive to avoid post-hoc explanation of patterns found in the models, the important variables, on the whole, match intuition relatively closely as to key drivers of health. The most important variable, by far, was Median Household Income, long considered a key driver of healthcare inequalities by experts[3]. Other key tweet topics that hewed closely to intuition were tweet frequency from topic 501, which concerns coming home and the associated positive feelings with one's home, which would occur more frequently in areas where people living at home are healthy. Additionally, tweet frequency concerning homework and school along with crying and anger also were important, each of which has easily understood predictive power on health outcomes. Though there were some puzzling inclusions in this summary of variable importance, like topic 584, with top words "Alex", "Jasmine", and "Walked", these variables do paint a strong picture of health predictors. Income, schooling, ability to travel and explore, and enjoying your home will lead to improved healthcare outcomes, on average. Crying and upset, on the other hand, are linked to poorer health statistics. Comprehensive study of these important predictors has key policy and human-level impacts, and should be considered further in healthcare interventions.

Conclusion

Ultimately, the model results provided fascinating information about both the state of healthcare in the U.S and the advantages and limitations of different modeling approaches in modern machine learning. The elastic net method proved best, far outpacing our attempt at a multi-task learning elastic net, which fared the worst. Though this may be counter-intuitive theoretically, as there were certainly unexplained correlations between outcomes in the elastic net model that a multi-task model would ostensibly work to correct, the fault lay primarily in mechanics. Without a full study or understanding of multi-task learning, an extremely active area of research in machine learning today, we were unable to effectively account for the latent similarities between outcome variables. Our PCA-based approach on the weight vectors in the elastic net was fruitless, indicating a different approach, loss function, or human input would be necessary to make this approach work. In this case, the ”KISS” approach - Keep It Simple, Stupid - was the winning strategy.

A few drivers of accuracy improvement included taking logs of the tweet topic probability in the elastic net, which allowed for a more coherent understanding of probabilities in log-space. And cross-validation of hyperparameters in every model proved extremely useful, showing the importance of fine-tuning even the best theoretically valid models. We did attempt some fruitless feature engineering approaches, including creating features that were linear combinations of indicators that already existed in the data. In hindsight, this information was already captured by the inputs, and more effective feature engineering would have focused on adding some level of novel information into the model, instead of relationships that were already accounted for.

Going forward, the biggest improvement we would make is our approach to multi-task learning. One idea is to create intelligent clusters on the given training Y's, then compare each predicted outcome with the cluster centers for the given Y's. We would then look for discrepancies, i.e. a county that should be in the ”poor health” group but with predicted obesity rate that is too low - and adjust the predicted outcomes towards the clustered centers. This would account for some level of correlation across outcomes. Additionally, as always, new predictors such as local hospital and doctor data could provide valuable. A superior multi-task approach could lead to more accurate predictions, which means safer, healthier, and happier people across the country.

References

[1] Sawyer, Bradley and Samarin, Alexander. *Peterson-Kaiser Health System Tracker.*

[2] Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. *Elements of Stastistical Learning.* Springer, 2017.

[3] Ruvolo, Paul and Eaton, Eric. *ELLA: An Efficient Lifelong Learning Algorithm.* `https://www.seas.upenn.edu/ eeaton/papers/Ruvolo2013ELLA.pdf`.