CIS 520 Final Project Report

Machine Hunter (Wenbo Zhang, Yufu Wang, Zheyuan Xie*)

Fall 2018

1 Introduction

In the final project of CIS 520: Machine Learning course, we developed a machine learning system to predict health behaviors and outcomes for US counties based on:

- 6 Demographics features (age, gender, race, etc.)
- 9 Socio-Economic Status (SES) features
- 6 Urban/rural environment features
- 2000 LDA topics in tweets from that county

We were provided with a training set of 1019 labeled training samples, each contains 6 demographics features, 6 urban/rural environment features, 12 SES features, and 2000 word features. We submit our models to the autograder which gives us as feedback the test error on a validation set with 1000 samples. The size constraint of the model is $\leq$ 50Mb, and the run time constraint of prediction is $\leq$ 10 min. Figure 1 shows all of our leaderboard submission results. We passed the Baseline 1 (0.1479) at the first attempt and passed the Baseline 2 (0.0810) quickly with a second submission. **Our final submission to leaderboard achieved a test error of 0.0726 on the validation set**, ranked 6th out of 66 teams in class.

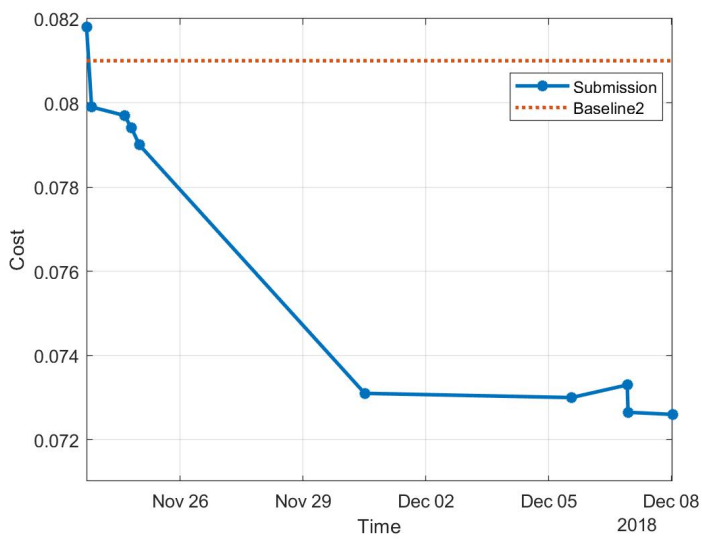


Figure 1: A time plot of our leaderboard submissions showing our progress.

In the following sections we will present an analysis of the methods we tried, and the cross-validation error of each methods. Based on our model, We also did some investigation that find out some interesting facts about public health in the United States.

*Names listed in alphabetical order

2 Methods

2.1 Overview

We explored a variety of methods to learn the dataset. Through applying feature selection and generation to the first 21 columns in the training data, we filter out less important features, reveal important features, and generated nonlinear features for the model. We also apply PCA to the last 2000 columns to reduce dimensions. Afterwards, we investigate the model performance through trying various regression methods, such as linear regression, elastic net, multi-layer perceptron, random forest, K-nearest neighbour, etc. The Train, Cross-Validation, and Test error of these methods are reported in Table 1.

Table 1: Experiment Result

Method	Train Error	CV Error	Test Error
PCA + OLR	0.0706	0.0784	0.0810
PCA + Elastic Net	0.0672	0.0777	0.0790
PCA + MLP	0.0806	0.0891	\
PCA + Gaussian Process	0.0354	0.0708	0.0727
Random Forest + OLR	0.0747	0.0871	\
10-Nearest-Neighbor	\	0.1166	\

* Train error not available for instance-based methods.
** Test error not available for methods not submitted to leaderboard.

Followed are our training procedures and the analysis of each method with their errors. Most importantly, our best model is Gaussian Process Regression with it novel Matern covariance Function. We consider each method in turn.

2.2 Dimensionality Reduction Method

2.2.1 Feature Selection

We adopted stage-wise regression as a feature selection process on the first 21 demographic/socioeconomic features. By adding in these feature one by one into an ordinary regression model, features were selected according to their contribution to the improvement of prediction accuracy. Some features were left out due to stagnation of their contribution. Through this process, it revels that features such as percentage of female, college education, food environment, household income and income ratio are the top 5 important factors in predicting the general health (for all 9 labels; not specific to one) of a county. We also find redundent features such as `mediummetro` and `smallmetro`, as we already included many other related features such as `largemetro`, `metrononmetro` and `nchs_2013`.

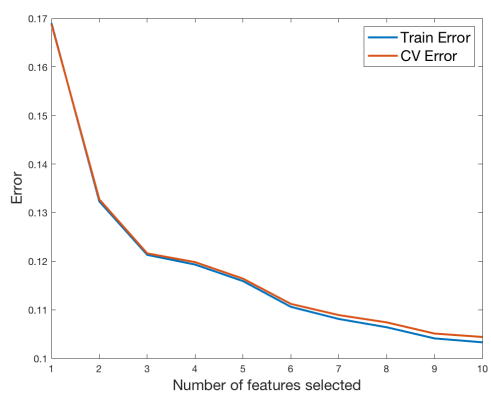


Figure 2: The effect of selected features on Error

We also used the same procedure to generate non-linear features such as square or square root of some of these features to enable the model to learn features nonlinear to the input. We found that square features such as household income, percentage of Hispanic, college education and access to healthy food can contribute to improvement. Square root of percentage of white population and people over 65 also were shown to improve model prediction.

There selected and generated features were used, along with the principal components of the 2000 LDA topics, in all of our models.

2.2.2 PCA

We used Principal Component Analysis (PCA) for dimensionality reduction of the 2000 LDA topics. Figure 3 shows percentage of variance explained by the principle components, the first four principal components explained most (72.7%) of the variance.

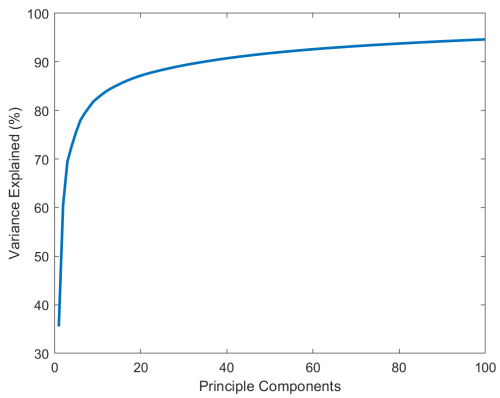


Figure 3: Percentage of variance explained by the first 100 PCs.

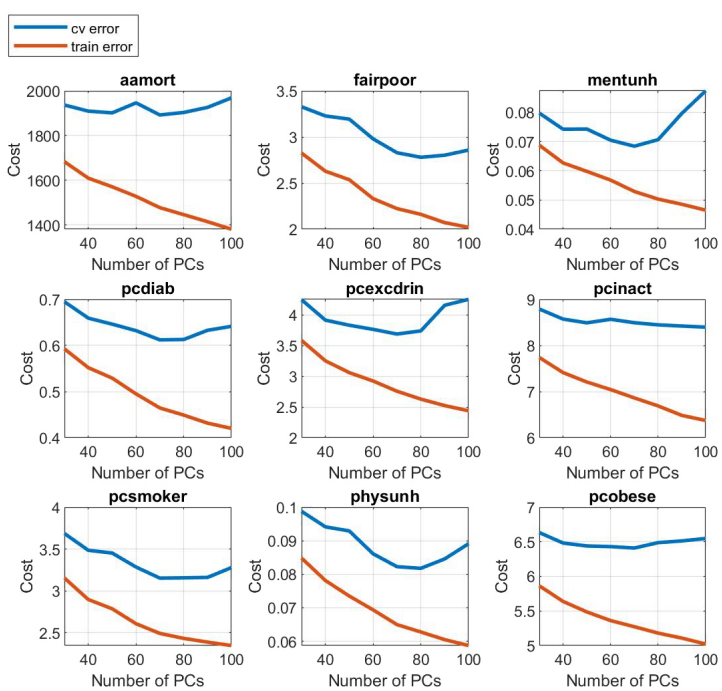


Figure 4: Effect on regression errors with number of principle components used.

To determine the number of principal components for training, we performed 5-fold cross-validation for each output y_i, using a simple linear regression model. We use the first n principal components ($n = 30, 40, 50, 06, 70, 80, 90, 100$) and then plot the cross-validation errors in Figure 4. For most of the outputs (except y_6), we have the best result when $n = 70 \sim 80$. The first 80 PCs explains 93.8% the variance, while $81 \sim 2000$ PCs explains only 6.2% which are probably noise and will cause the model to overfit.

2.2.3 Random Forest

Instead of using random forest as an end-to-end regression model, we explored random forest as a discriminative process to select important topics from the 2000 topics features. By doing this, we can compare its effectiveness in extracting important features from the 2000 topics to other methods such as PCA. A random forest regression model was first created using the 2000 topics to predict. Then the most important 70 topics in the model were picked, along with the demographic/socioeconomic features, to be used in an ordinary linear regression as final model.

2.3 Regression Method

There are two main problems in Machine Learning: regression problem and classification problem. We are required to predict health behaviors and outcomes for US countries based on 21 features and 2000 latent topics. This is not a classification problem but a regression problem because we are not supposed to separate the input space into different categories or decision boundaries. Followed we will introduce various methods we apply for regression.

2.3.1 Linear Regression

Linear regression can pave a strong foundation for further complex model analysis. Therefore, the first method we try is linear regression.

In MATLAB, there is a build-in function `regress` for linear regression model. We shall know this is a complex problem, so using linear regression alone cannot solve the problem and may cause overfit. Due to this reason, we add the penalty to the function, and we find there is another build-in function `lasso` which can perform both lasso (α=0.5) or elastic net ($\alpha \neq$0.5) regularization for linear models.

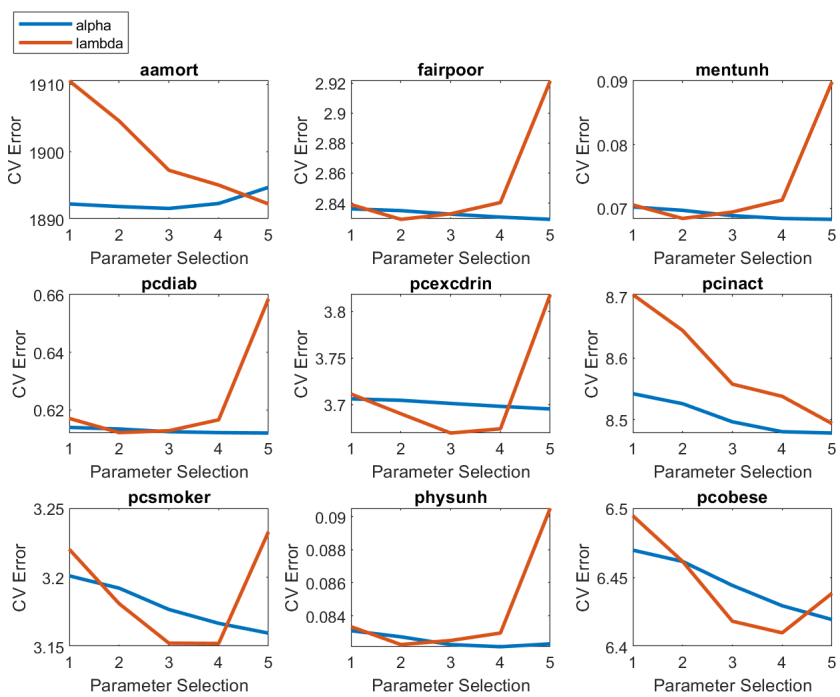


Figure 5: Parameter selection of elastic net by cross-validation

$$\min_{w,b}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^T w + b)^2 + \lambda\left(\frac{(1-\alpha)}{2}||w||_2^2 + \alpha||w||_1\right)\right)$$

For each label, we performed cross-validation to determine the elastic net parameter λ and α. We tested 5 different λ and α, a total of 25 possible combinations for each label. The cross-validation errors are reported in Figure 5. By choosing optimal hyper-parameters, we reduced the test error from 0.0799 to 0.0790. This slight improvement is also reflected in our leaderboard submission curve in Figure 1. Our final linear regression mdoel on PCA data achievesa test error of 0.0811, and elastic net with optimally chosen hyper-parameters achieves test error to 0.0790.

2.3.2 K-Nearest-Neighbour

We also implemented an instance-based method to solve the regression problem. For each new data point, we applied KNN to find the closest K points, and took the average as our prediction. Below is a figure showing the effect of K on cross-validation error and training error. From Figure 15, we get the insights that 10-nearest-neighbour produces the most accurate prediction. We can also find the trend in Figure 5: When K is small, the model is complex and the training error is small. As K increases, the training error also goes up because the model becomes simple and 'smooth'. As for cross-validation error, the trend of the curve decreases tremendously for the first part and then increases slowly. Because the model is overfitting when K is small. After the model achieves it optimal K when $K = 10$, the error increases since the model becomes simple and cannot reflect the distribution of the whole data set. However, KNN only achieves 0.1166 cross-validation error even when K is selected to be optimal, way higher than other methods. **Normalizing the features or add scale factors to each feature may improve the result**.

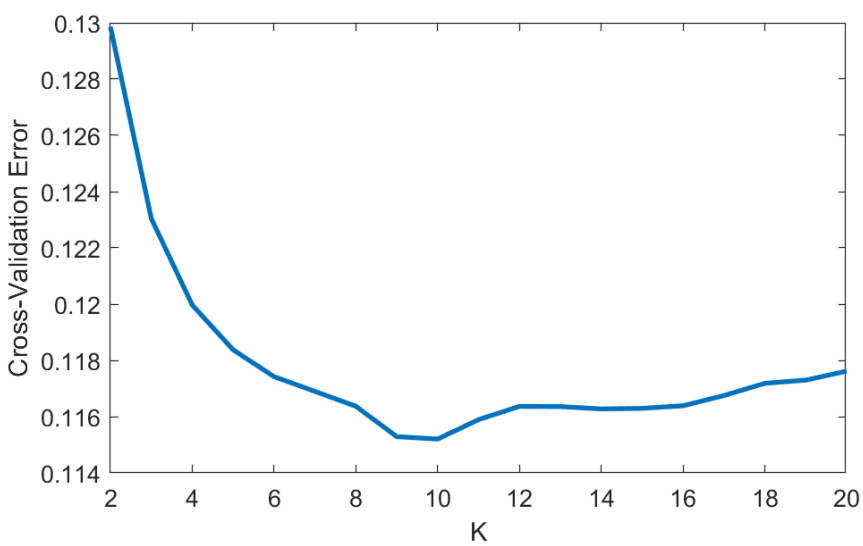


Figure 6: Cross-validation error with different K.

2.3.3 Multi-Layer Perceptron

We also experimented with multi-layer perceptron (shallow neural network), though we abandon this method very early on because of its tendency to overfit. For a two hidden layer neural network, each with 5 units, with 100 features for each input data point, we would have ∼600 parameters. Compared to the ∼1000 training data size, this model would need some serious regularization. We also tested with different activation functions (Sigmoid, Relu, tanh, etc.), however, none of them achieves a cross-validation error that is comparable to linear regression. Therefore, we decided to pursuit simpler model.

2.3.4 Gaussian Process Regression

The prediction performance of a Gaussian process model will partly depends on the covariance function we choose. Therefore, there are a lot of hyperparameters related to the covariance function we can use, such as the length scale of the correlations, the precision of the noise and so on.

In MATLAB, there are ten build-in kernels to form the covariance functions, including exponential kernel, squared exponential kernel, matern kernel with parameter 3/2, matern kernel with parameter 5/2, rational quadratic kernel, exponential kernel with a separate length scale per predictor, squared exponential kernel with a separate length scale per predictor, matern kernel with parameter 3/2 and a separate length scale per predictor, matern kernel with parameter 5/2 and a separate length scale per predictor, rational quadratic kernel with a separate length scale per predictor. Because this is one of the most important parameters we shall investigate, we set all the other parameters into default mode and tune 'KernelFunction' in the first step.

The mathematical equations for these kernels are as followed. In these kernels, x_i is predictor values, y_i is response (target) values, σ_f is standard deviation, σ_l is characteristic length scale, θ is the unconstrained parameterization vector which can enforce the value of σ_l and σ_f to be greater than 0, such that

$$\theta_1 = \log \sigma_l,\ \theta_2 = \log \sigma_f$$

- Exponential Kernel

$$k(x_i, x_j|\theta) = \sigma_f^2 \exp(-\frac{r}{\sigma_l})$$

  where $r = \sqrt{(x_i - x_j)^T(x_i - x_j)}$.
- Squared Exponential Kernel

$$k(x_i, x_j|\theta) = \sigma_f^2 \exp[-\frac{1}{2}\frac{(x_i - x_j)^T(x_i - x_j)}{\sigma_l^2}]$$

- Matern 3/2

$$k(x_i, x_j|\theta) = \sigma_f^2\Big(1 + \frac{\sqrt{3}r}{\sigma_l}\Big)\exp\Big(-\frac{\sqrt{3}r}{\sigma_l}\Big)$$

  where $r = \sqrt{(x_i - x_j)^T(x_i - x_j)}$.
- Matern 5/2

$$k(x_i, x_j|\theta) = \sigma_f^2\Big(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\Big)\exp\Big(-\frac{\sqrt{5}r}{\sigma_l}\Big)$$

  where $r = \sqrt{(x_i - x_j)^T(x_i - x_j)}$.
- Rational Quadratic Kernel

$$k(x_i, x_j|\theta) = \sigma_f^2\Big(1 + \frac{r^2}{2\alpha\sigma_l^2}\Big)^{-\alpha}$$

  where $r = \sqrt{(x_i - x_j)^T(x_i - x_j)}$ and α is a positive-valued scale-mixture parameter.

Through only changing this parameter, we achieve different kernel's influence on the 5-fold cross-validation error. From Table 2, we shall clearly find the Matern 3/2 works best. Therefore, we set this parameter 'matern32' to be unchanged and further investigate the other parameters' influence on the results. Because Kernel functions with separate length scales for predictors usually need more than 10-minute average training time, we consider not to use these kernel functions when we have a requirement of less than 10-minute training time.

Besides 'KernelFunction' mentioned above, we also change 'FitMethod' and 'BasisFunction'. 'FitMethod' uses a method to estimate parameters of the Gaussian Process Regression model. In order to grasp a better

Table 2: Training and Cross-Validation Errors v.s. Kernel Functions

Kernel Function	matern32	matern52	rationalquadratic	exponential	squaredexponential
Training Error	0.0354	0.0413	0.0396	0.0030	0.0468
Validation Error	0.0709	0.0714	0.0713	0.0712	0.0724

Table 3: Training and Cross-Validation Errors v.s. Fitting Functions

Fitting Method	none	fic	exact	sd	sr
Training Error	0.4449	0.0354	0.0354	0.0354	0.0354
Validation Error	0.8900	0.0709	0.0709	0.0709	0.0709

sense of which specific fitting method works best, we plot the cross-validation error with different fitting methods.

In the Table 3, 'none' is no estimation, 'exact' is exact gaussian process regression, 'sd' is subset of data points approximation, 'sr' is subset of regressors approximation, 'fic' is fully independent conditional approximation. From the result in the Table 3, we learn all the methods except 'none' all have the same impact on the training and validation error. Since the number of our training data is less than 2000, we would like to use 'exact' mode as it is a default model if $n \leq 2000$ (n is the number of observations).

As for 'BasisFunction', there are five options ('none', 'constant', 'linear', 'pureQuadratic', 'hfcn'). According to MATLAB document, the basis function adds the term $H * \beta$ to the model, where H is the basis matrix and β is a p-by-1 vector of basis coefficients. When we apply 'linear', 'pureQuadratic' and 'hfcn', there is an error that the matrix is not of full column rank. So we choose 'constant' for this hyperparameter.

After tuning the hyperparameters, we set 'KernelFunction', 'FitMethod', and 'BasisFunction' to be 'matern32', 'exact', and 'constant', respectively. Finally, we use 5-fold to calculate its cross-validation error and submit them on the leaderboard. This is the best model we got, which achieves 0.0354 train error, 0.0708 cv error, and 0.0727 test error.

2.4 Leaderboard Submission

We submit three models on the leaderboard, which are based on `PCA+OLR` (Ordinary Linear Regression), `PCA+Elastic Net`, and `PCA+GPR` (Gaussian Process Regression). We repeatedly submit to the leaderboard and constantly tuning parameters to make the result better. Among the three models we attempt to submit to leaderboard, `PCA+GPR` has the best performance because the first two models are linear regression but the third model is Gaussian process regression. Nonlinear model can fit into high-dimensional dataset and present its distribution well. Followed paragraphs introduce our submission procedure in detail.

Firstly, as mentioned above, we do the feature selection and PCA to the first 21 features (Demographics, Socio-Economic Status, and Urban/Rural Environment) and 2000 LDA tweet topics, respectively (Figure 2 and Figure 4). This method performs fairly well in reducing the dimensionality of training data and selecting useful features.

In order to beat the first baseline, we apply a simple model based on linear regression. Because linear regression has good analytical properties and paves a strong foundation for further sophisticated model analysis [1], we use it first to provide us with a benchmark for this project.

Considering the limitations of linear regression model for its lack of practical techniques for solving problems involving high dimensional input space, we put an eye on the non-linear regression models after we cannot improve the performance of the linear model. Gaussian Process Regression is a good method for us to have a try because it is not only non-linear method but we can change the kernel function of it to adjust different nonlinearities.

Although we have not tested emsemble methods of integrating the advantages of different methods with good performance (e.g. give weights w_i to the output of the function $pred_labels$ and normalize the output), our `PCA+GPR` can achieve top 6 before the winner prize deadline. It proves the simple and single model can also perform pretty well if we tuned the parameters in a correct way.

3 Data Interpretation

From the feature selection process, we find that some demographic and socioeconomic features are more important than others to predict county health. Female percentage is shown to be an important indicator, and is the first feature being selected in a stage-wise process. Female statistically live longer [2]; they are more likely to be obese than male[3]; there are also many other health indicators associated with gender. So it makes sense that female percentage will play an important part in our model.

Other important factors include college education, food environment, household income and income ratio. This is corresponding to our understanding that a county with more educated and higher income (less income inequality) population tends to be healthier, since the population will be able and willing to afford better food, medical insurance, and clinical care.

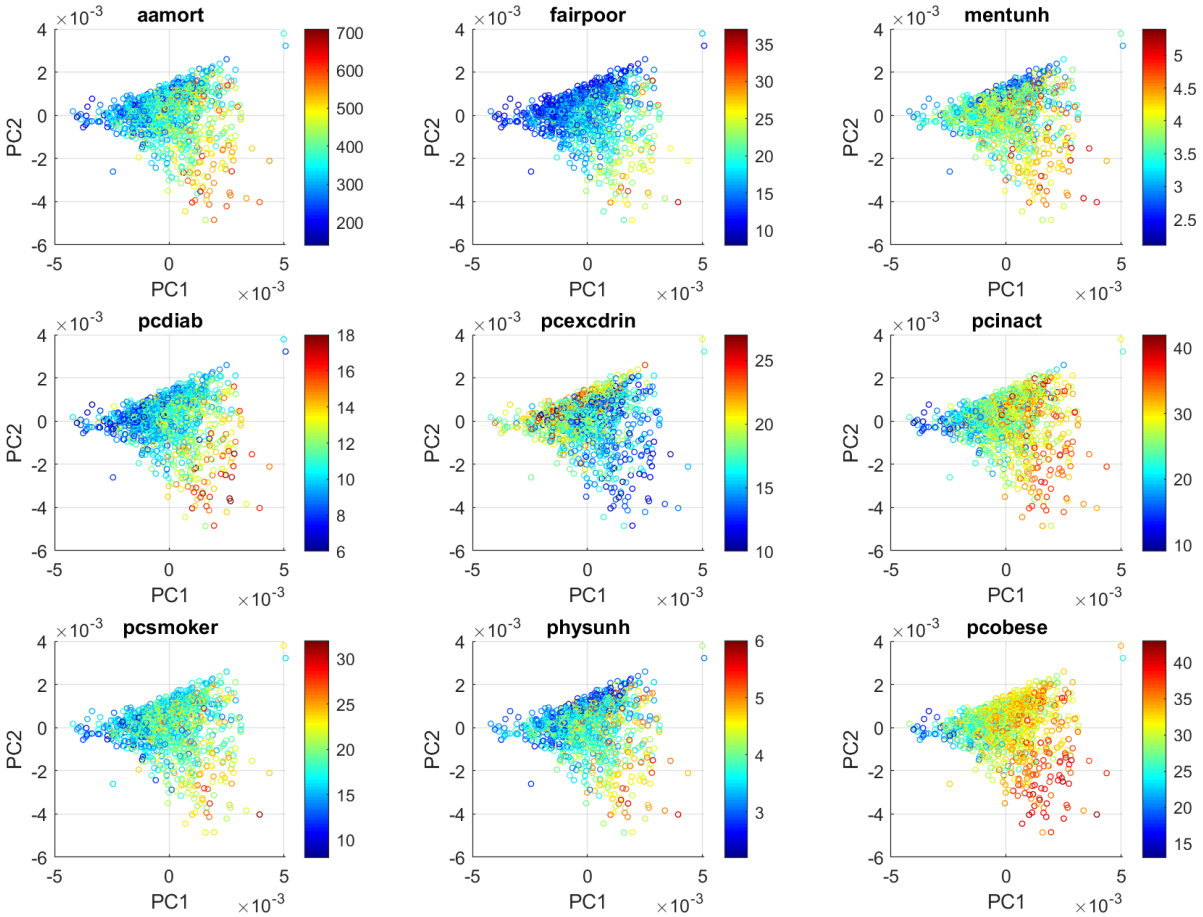


Figure 7: Training samples projected to the first 2 PC dimensions.

Our major treatment of the 2000 topic features is applying PCA as dimensional reduction (also as a mean of regularization). Each projection is a linear combination of topics, which can be interpret as a 'hypertopic'. Figure 7 shows data points for each of the 9 labels projected to PC2-PC1 plane. The values of PC1 and PC2 are scores of each data points. Color bar shows the the value for y labels of each data point. From Figure 7, high value (red color) are visually separated from data points with lower value (blue color). This implies that PC1 and PC2 are good indicators for county health. For all labels except `pcexcdrin` (excessive drinking), higher values in PC1 indicate lower county health (lower county health stands for higher `aamort`, `fairpoor`, `mentunh`, `pcdiab`, `pcinact`, `pcsmoker`, `physunh`, `pcobese`) and vice versa. So we can find out what topics PC1 is composed of, pick the ones with high positive and negative share , and visualize the wordcloud for those topics.

Figure 8: Topics with high values in PC1 (indicate low county health)

Figure 9: Topics with low values in PC1 (indicate high county health)

From Figure 8∼9, we see that PC1 could be an indicator of education and skill level. Word such as "company", "business", "engineering" and "meeting" are correlated with higher county health, while miss spelling and curse words are correlated with lower county health. However, PC1 also contains many topics that have high weights but do not clearly related to health issue. This aligns with the unsupervised nature of PCA: PCA capture the variance; not all variance are related to our subjects.

To get more insights on county health from topics after PCA, we calculated the linear correlation between the score of a principle component and the 9 data label. We found that some of the principal components (1,2,7,9,16,17,29,39) is well-related to health while others are not even if explains a high percentage of variance (3,4,6,8). For each of the 9 labels, we projected the data points to their first two most correlated principle components (MCPC), as shown in Figure 10.

MCPC / Label	1		2		3		4	
	PC	Corr	PC	Corr	PC	Corr	PC	Corr
aamort	1	**0.50**	7	-0.34	9	0.27	2	-0.25
fairpoor	1	0.48	2	-0.38	9	0.30	7	-0.26
mentunh	9	0.31	7	-0.31	2	-0.27	17	-0.25
pcdiab	1	**0.53**	2	-0.34	9	0.26	29	-0.20
pcexcdrin	1	-0.37	9	-0.34	7	0.34	2	0.21
pcinact	1	**0.60**	7	-0.32	9	0.21	5	-0.18
pcsmoker	1	0.40	7	-0.29	17	-0.25	9	0.18
physunh	1	0.35	9	0.35	7	-0.32	2	-0.26
pcobese	1	**0.61**	7	-0.21	4	-0.19	2	-0.16

Table 4: Most correlated principle components for each label

PC	High1	High2	High3	High4	High5
1	bit,:/,ly	follow,twitter,tumblr	az,meg,rt	shut,ass,bitches	da,dat,wit
2	bit,:/,ly	az,meg,rt	ne,ut,sh	follow,twitter,tumblr	you're,you'll,you've
7	de,je,le	bit,:/,ly	fuck,shit,fucking	shit,ass,bitches	fuck,fucking,shit
9	de,je,le	photos,pictures,pics	god,jesus,christ	god,lord,praise	church,sunday,service
17	north,south	back,florida,fl	justin,bieber,beiber	(:,text,haha	texas,houston,austin

Table 5: Most frequent words in health-related PCs

PC	Low1	Low2	Low3	Low4	Low5
1	photos,picture	vote,president	pm,free,saturday	government,economy	video,youtube
2	shit,ass,bitches	da,dat,wit	nigga,aint,ya	dnt,wat,ur	yu,dnt,kno
7	follow,twitter,tumblr	justin,bieber,beiber	god,jesus,christ	god,lord,praise	church,sunday
9	follow,twitter,tumblr	state,ohio,michigan	de,te,el	justin,bieber,beiber	lead,path,walk
17	vidoe,youtube,video	mountain,mt,dew	colorado,heading	follow,twitter,tumblr	lol,jk,lmao

Table 6: Least frequent words in health-related PCs

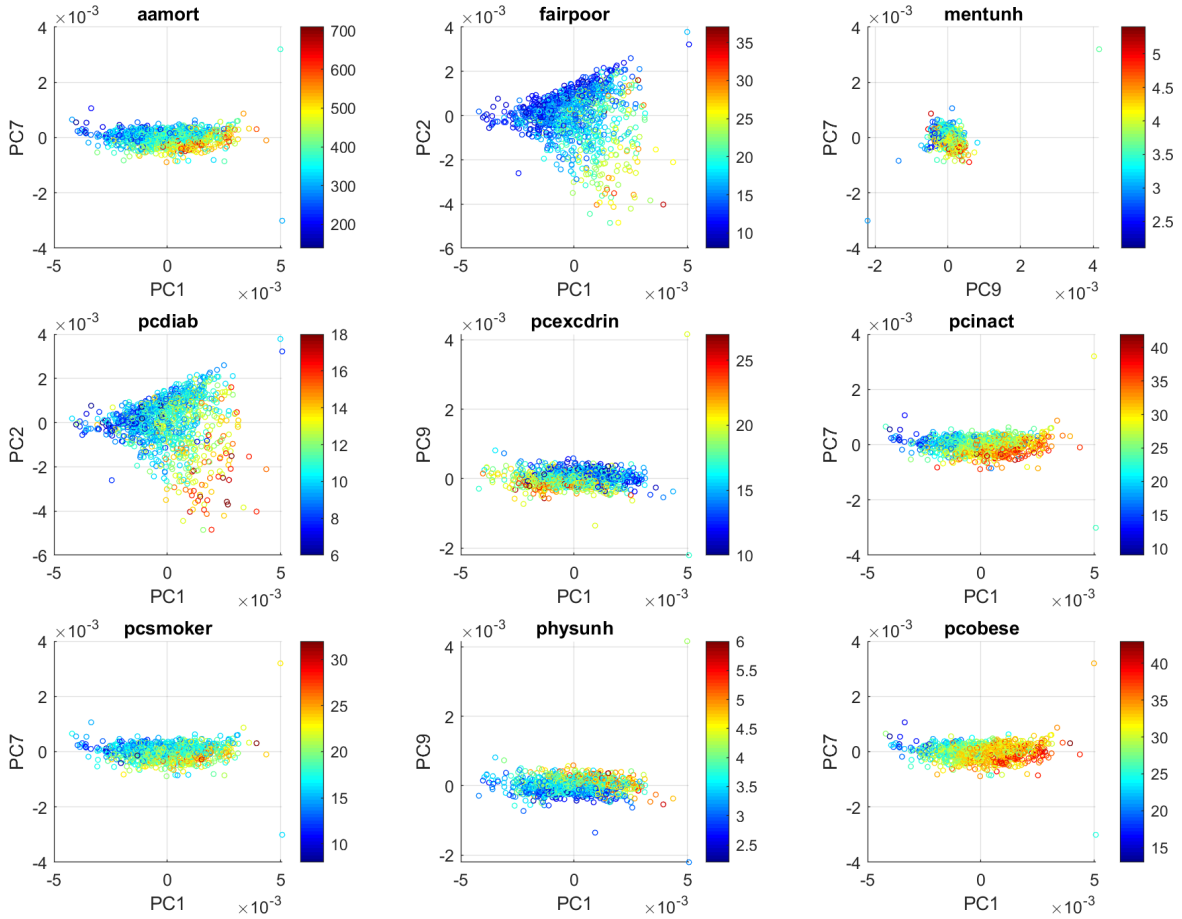


Figure 10: Training samples projected to the most correlated 2 PC dimensions.

Some of the facts we can infer from Figure 10 together with Table 5 and 6 is:

- The mental health in a county has little to do with tweet topics, probably because people with mental problem tweets less frequently than people without mental health problem.

- The 1st principle component is the most predictive about county health. This is intuitively well-explained that counties with better education and higher income have healthier people. The scores of PC1 is visualized in Figure 11 on a US county map.
- The 7th principle component is related to `aamort`, `mentunh`, `pcinact`, `pcsmoker` and `pcobese`. Almost all the topics it positively correlate to consists of f-words.
- The 9th principle component is related to `pcexcdrin` and `physunh`. Interestingly, people in counties that report more excessive drinking tends to be physically healthier. They also tweets more on religion (god, Jesus, church, etc.) but not in the Great Lakes region (Ohio, Michigan).
- It is indicated in many ways that counties that tweets more slang words are poor in health.
- Justin Bieber appears in many correlated topics and has a strong correlation with Christian religion. We look into this by looking for geography map about Justin Bieber fever and Christian population. As Figure 14 indicates, both topics might suggests some information about the state in which county is located. We drew PC7 and PC9 scores on a US county map as shown in Figure 12 and Figure 13, we can see that regional difference is apparent.

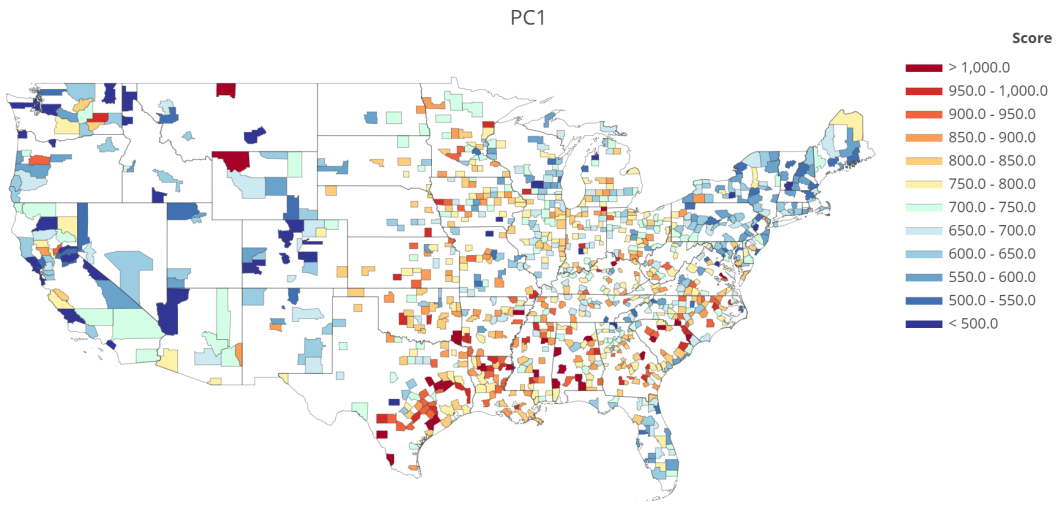


Figure 11: Scores of PC1 in training set.

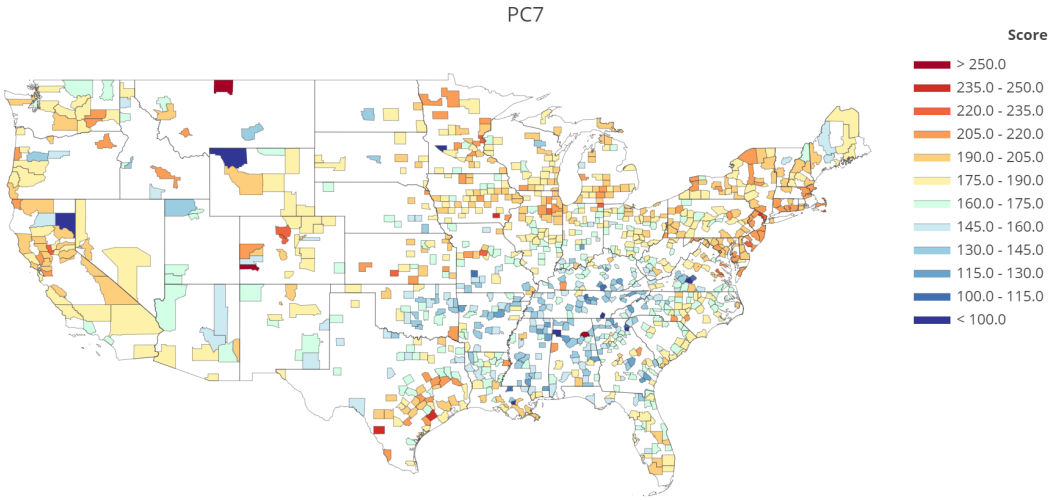


Figure 12: Scores of PC7 in training set.

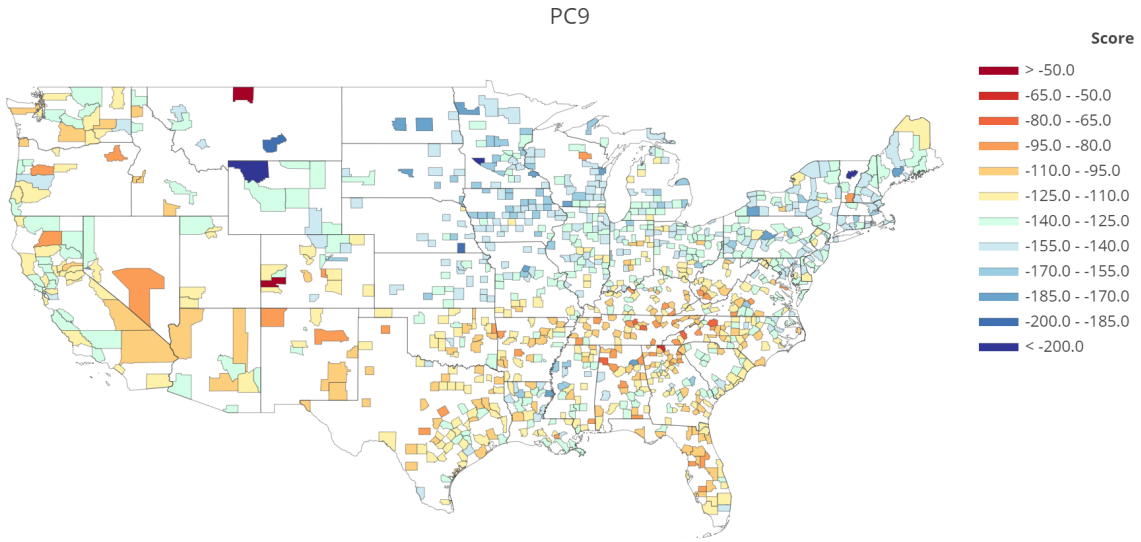
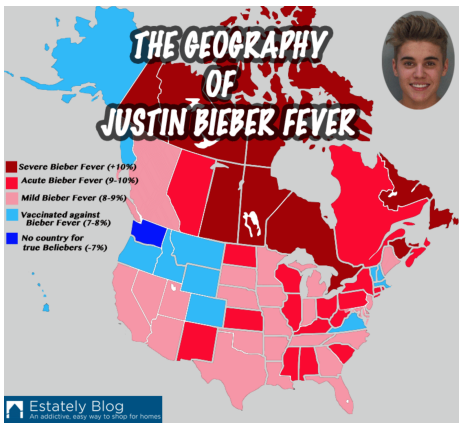
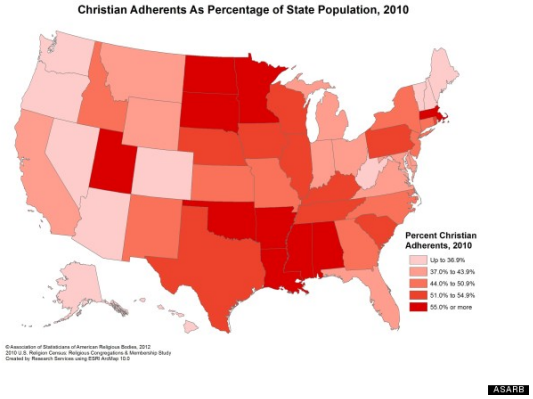


Figure 13: Scores of PC9 in training set.



(a) Justin Biber fever [4]



(b) Christian Population [5]

Figure 14: Geography of topics.

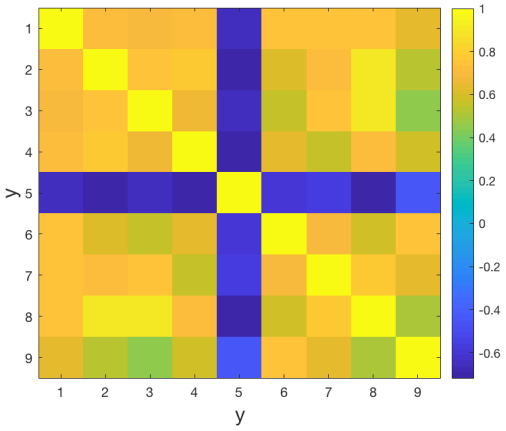


Figure 15: Label correlation

Last but not least, we investigated the correlations between the 9 labels. As seen in Figure 12, most labels are positively correlated; this accords to our common sense that counties that are unhealthy in one measures tend to be also unhealthy in other measure. One exception is label 5, percentage of adults reporting excessive drinking. One interpretation is that wealthier counties can afford more alcohol and tend to engage in more parties so alcohol consumption in these counties can be higher. Another hypothesis is that excessive alcohol consumption actually is good for health. This definitely deserves more investigation.

4 Conclusion

4.1 Model Analysis

Among all the regression models we experimented, **PCA+GPR** (Gaussian Process Regression model) works the best and **K-nearest neighbour regression** works the worst. Figure 16 shows the trends in errors of all our models.

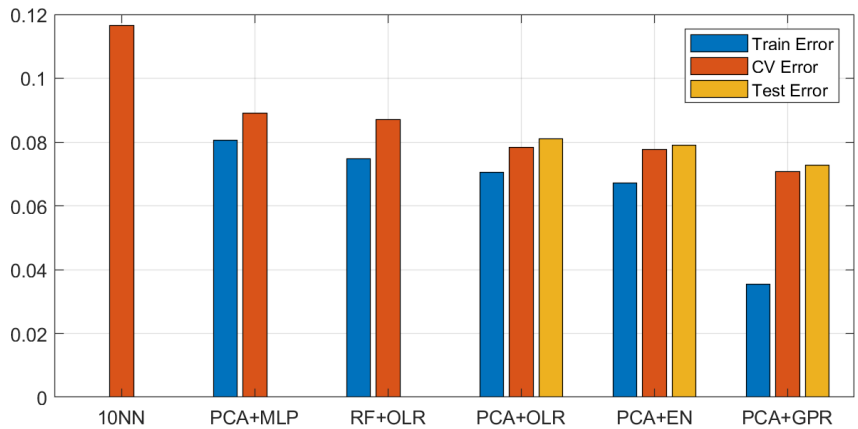


Figure 16: Performance across methods

To improve the accuracy of prediction, we tried the following methods:

- Dimensionality reduction: Use PCA and random forest to reduce 2000 LDA features and apply feature selection to reduce 21 features (Demographics, Socio-Economic Status, and Urban/Rural Environment).
- Feature engineering: Feature creation and feature transformation are feature engineering methods. Through adding another columns of square root and square of the original data, we increase the feature dimension for our input data set.
- Model selection: Try different regression models and penalize cost function.
- Linearity to non-linearity: Add non-linear features, use non-linear regression methods.
- Hyperparameters optimization: Use cross-validation to select best hyperparameters.

Besides, we also faced some problems when the results were not matched with our expectation:

- In Gaussian Process Regression, there are several kernel functions (mentioned in 2.3.4) we can use, such as matern covariance function, exponential function, squared exponential function, and so on. However, when we gave weights to the results of different Gaussian process models with their kernel functions, the cross-validation error is higher than the model with only matern covariance function. The ensemble method does not work in this situation. We think the reason is matern covariance function works best for the whole dataset. Although we integrated other kernel functions, their performances are worse than 'matern32'. As a result, ensemble operation hinders the good performance of 'matern32'.
- When we ensembled (average, weighted average) three models (OLR, SVM, Gaussian Process Regression) together and then used cross-validation, the performance of the ensembled model turn out to be worse than `PCA+GPR` alone. Through analyzing our result and comparing the ensemble model with others, we shall ensemble more 'weak' models to fully exploit its advantages.

Figure 16 shows non parametric method (KNN) performs worse than parametric method. This is because the curse of dimensionality. The number of examples that we have in a volume of space decreases exponentially with the number of dimensions. We have only 1019 training samples but 2021 features, even if we use PCA the number of features is still too many for instance-based method. Similarly, MLP performs worse then linear regression probably because there are too many parameters but too few training data, overfit is likely to happen. Another possible reason is MLP have nonlinear component, but we did not normalize the inputs. Elastic net performs slightly better than non-penalized linear regression since its L1 and L2 penalized term prevents overfit.

4.2 Improvement and Future Work

In the end, comparing our results with other top teams, we think there are three ways to further improve our model:

- Feature engineering: There are other feature creation method we could apply. Because we have already known the FIPS code of some counties in our training data, we can find the FIPS codes that are not in our training data, predict 'unknown' counties' feature manually and add them to the training data (latent variable). It could work fairly well due to our prior knowledge about these 'unknown' counties.
- Fancy function approximation: It is theoretically possible to use deep neural networks to approximate the dataset well since they are good at fitting functions. But it can be challenging to choose hyper-parameters, and may also need a bit more engineering such as normalizing the input data or adding dropout layers to prevent overfit.
- Ensemble learning: Although we applied ensemble learning and the cross-validation error were higher than the non-ensemble model, we can integrate more models to make it better.

References

[1] Christopher M Bishop. "Pattern recognition and machine learning (information science and statistics) springer-verlag new york". In: *Inc. Secaucus, NJ, USA* (2006).

[2] Steven N Austad. "Why women live longer than men: sex differences in longevity". In: *Gender medicine* 3.2 (2006), pp. 79–92.

[3] Rebecca Kanter and Benjamin Caballero. "Global gender disparities in obesity: a review". In: *Advances in nutrition* 3.4 (2012), pp. 491–498.

[4] Ryan Nickum. *The Geography of Justin Bieber Fever*. Blog. 2014.

[5] Clifford Anthony Grammich. *2010 US Religion Census: Religious Congregations & Membership Study: an Enumeration by Nation, State, and County Based on Data Reported for 236 Religious Groups*. Association of Statisticians of American Religious Bodies, 2012.