Exploring and Predicting US Health Insurance Premiums

PIC 16B Project Report

Group: Dakota Lin, Anshika Khandelwal, Sarah Ward

1. Introduction

1.1 About the Dataset

The “Insurance Dataset for Predicting Health Insurance Premiums in the US” from Kaggle is a comprehensive collection of data designed to explore factors affecting medical costs and health insurance premiums in the United States. It includes ten variables: age, gender, BMI, number of children, smoking status, region, income, education, occupation, and type of insurance plan. Generated by a script that created one million random data points, the dataset reflects the insured population in the US. This dataset is suitable for constructing and evaluating machine learning models to forecast insurance premiums and analyze the impact of various factors on medical expenses.

Outlined below are the details of the features of the dataset:

- **Age**: numerical variable. Average age in the dataset is 41.5 years, and the maximum age is 65.
- **Gender**: categorical variable. Dataset has an even split of males and females (50% each)
- **BMI**: numerical variable. Average BMI in the dataset is 34, with the highest value being 50.
- **Number of children**: numerical variable. Average number of children is 2.5, and the maximum number of children is 5.
- **Smoking status**: categorical variable. Options are True or False.
- **Region**: categorical variable. Regions include Northeast, Southeast, Northwest and Southwest, as the dataset is limited to the US.
- **Medical history**: categorical variable. Options include heart disease, high blood pressure and diabetes.
- **Family medical history**: categorical variable. Options include heart disease, high blood pressure and diabetes.
- **Exercise frequency**: categorical variable. Options include rarely and occasionally.
- **Occupation**: categorical variable. Options include blue-collar, white-collar, student and unemployed.
- **Coverage level**: categorical variable. Options include basic, standard and premium.
- **Charges**: numerical variable. This is our target variable.

1.2 Project Goals

Now that we have a better understanding of the dataset, our goals can be better understood. Primarily, our goals with the dataset are to:

- **Explore** and visualize the **correlations between various factors** that could affect health premiums
- Find the best model that would be able to **predict insurance charges** for individuals based on important features from the dataset.

2. Data Preprocessing and Preparation for Training

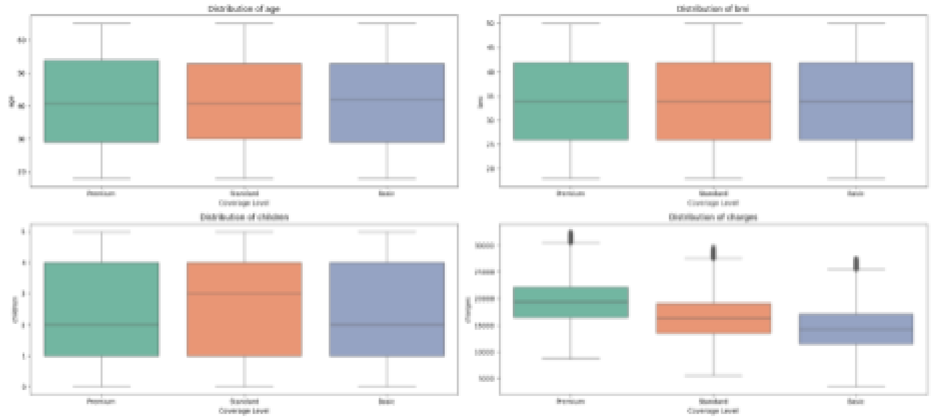
The main preprocessing steps we took are as listed as follows:

1. There are null values found in medical_history and family_medical_history fields. These entries are kept and replaced with string "None" because having no medical history is also an important factor.
2. We generated a smaller sample dataset of size 10,000 from the original data with random seed to reduce computation time.
3. We encoded the 8 categorical features with one-hot encoding, which transformed our dataset to have a total of 30 features.
4. We standardized the 3 numerical features (age, BMI, and number of children) by defining a class for standardization.
5. Using train test split, we split the smaller sample dataset into a train set of size 8,000 and a test set of size 2,000.

3. Exploration of the Relations Between Variables

3.1 Feature Comparisons to Coverage Level

We began exploring our data by searching for any relationships between our variables through visualizations. First, we compared the numerical features of our data to the coverage level.

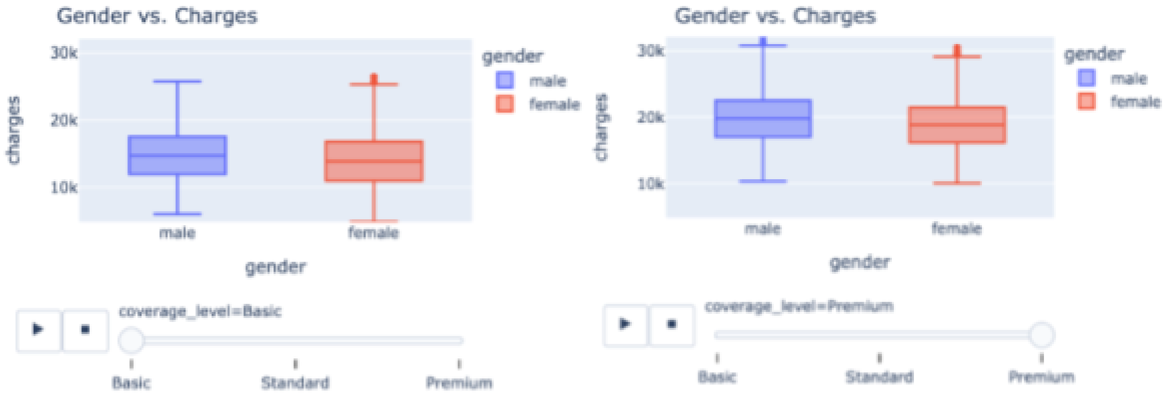


Plot showing the relationship between numerical features of age, bmi, children, and charges versus coverage level

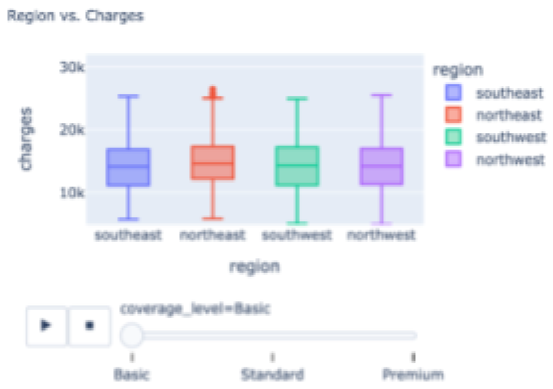
This comparison between the numerical features and coverage level isn't of much use to us moving forward as there aren't any clear indications of significant relationships that will help us toward our goal so we will look to other distributions in comparison to charge.

3.2 Feature Comparisons to Charge

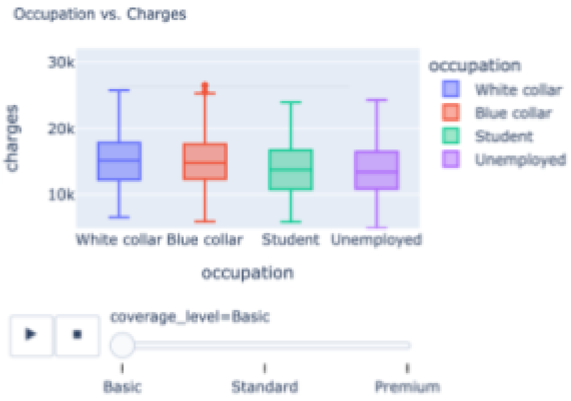
We want to look further into comparisons between various features and how much these individuals are being charged based upon those features so we created visualizations for various variables against charge.



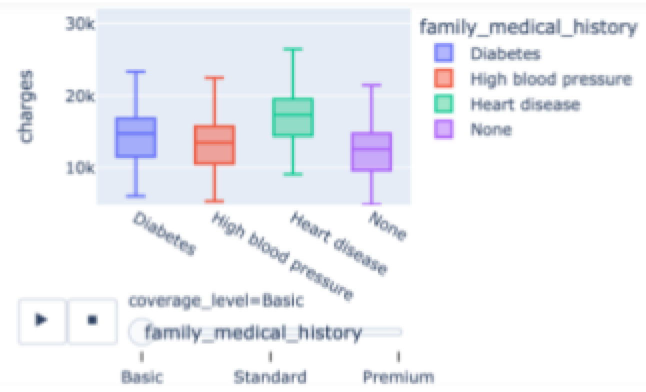
Plot showing the relationship between gender and charges for each coverage level



Plot showing the relationship between region and charges for each coverage level



Plot showing the relationship between occupation and charges for each coverage level

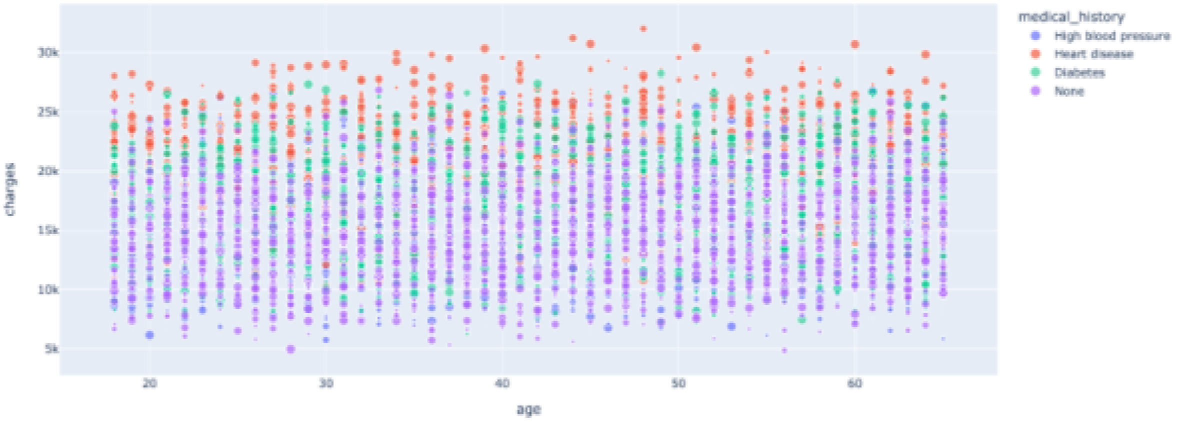


Plot showing the relationship between family medical history and charges for each coverage level

From the boxplots above, plotting all the categorical features vs the charges, we can see that:

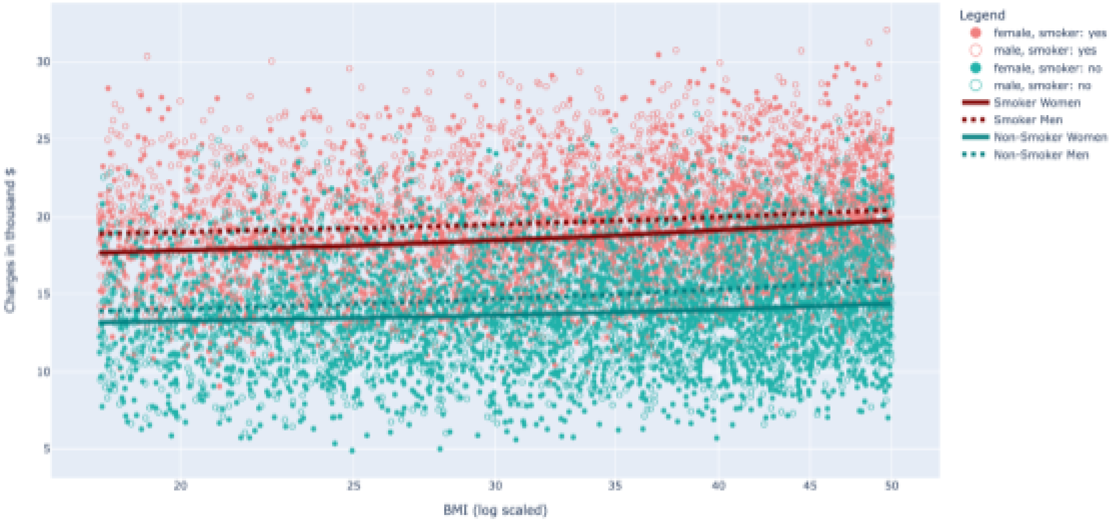
- Region does not have a large effect on charges. Regardless of whether the individual is from the southeast, northeast, southwest or the northwest, charges are relatively consistent.

- Gender does not have a large effect on charges. However, males have a slightly larger insurance charge than females.
- Smokers have significantly higher charges when compared to non-smokers
- People with no recorded medical history have the lowest charges, followed by those with high blood pressure, then diabetes. Those with heart disease histories have the highest charges. The same pattern is true for family medical history.
- Unemployed individuals and students have relatively similar charges, whereas people with white and blue collar occupations have relatively higher charges.



Plot showing the relationship between charges and age for each individual's medical history

We want to further explore relations between an individual's medical history and their age versus what they are being charged for their medical insurance plan so we look to the plot above. It seems that more individuals without a medical history are being charged less. This makes sense because you wouldn't want to be paying more for your health insurance if you don't have any health concerns to begin with. On the other hand, heart disease appears to be the major medical history factor responsible for high insurance charges, and diabetes follows. There isn't an apparent trend for difference in ages between individuals.



Plot showing the relationship between charges and BMI against smoking status and gender

BMI will likely be one of the most important numerical features as it is the only one that can be significantly correlated to health status. Smoking Status has been known as a significant cause of health conditions so it is likely an important variable in our data. Therefore we created a plot to visualize the comparison between BMI and premium charges for various smoking status within gender. Overall, by the scatter plot and regression lines, we notice that charges tend to go up as BMI increases for each insured individual. Charges are much higher for individuals who smoke compared to non-smokers. Men smokers tend to have higher charges than female smokers.

3.3 Supplementary Data Analysis: CMS National Health Expenditure

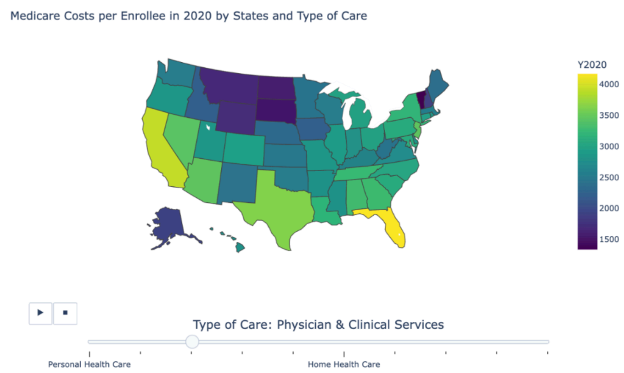
To verify and extend on the findings derived from our primary dataset, we incorporated an analysis of a supplementary dataset, the National Health Expenditure Data published by Centers for Medicare & Medicaid Services, as shown below in this section. Specifically, we retrieved the data that reflects the medical costs per Medicare enrollee by state of residence during 1991-2020. Using this data, we want to see if the interaction between factors affecting real-world medical costs aligns with the trends we discovered in our primary dataset.

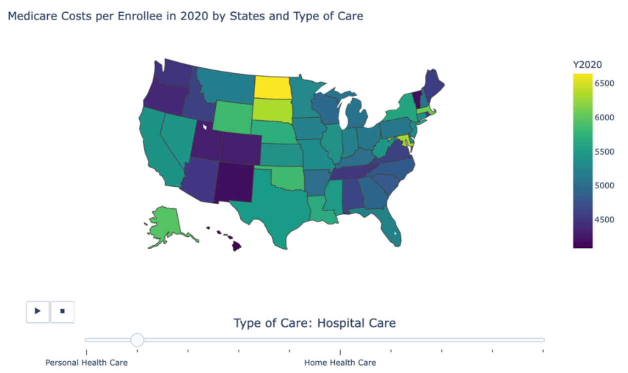
According to the official documentation provided by CMS.gov, the descriptions of the columns in this dataset are as follows:

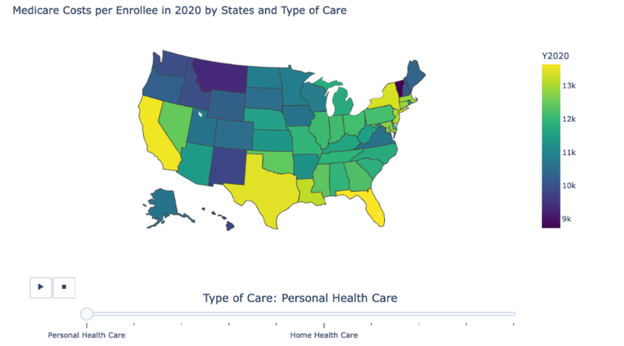
- **Code**: Numerical code assigned to each Item. Each number corresponds to a type of medical care
- **Item**: Identifies health spending level of aggregation/payer/service/good, enrollment, or population; appropriate units/scale (e.g. millions of dollars)
- **Group**: Level of aggregation by geography
- **Region_Number**: Numerical Code Assigned to each Region, for sorting purposes

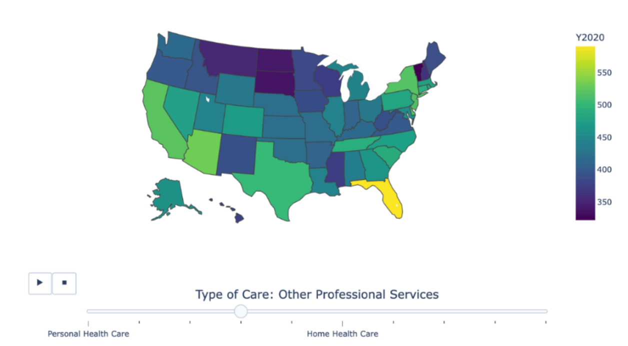
- **Region_Name**: Bureau of Economic Analysis Region Name
- **State_Name**: U.S. State Name
- **Y[xxxx]**: Spending for year [xxxx]
- **Average_Annual_Percent_ Growth**: Average annual growth rate for spending, 1991-2020

To more efficiently detect geographical information to be used for choropleth maps, we keep rows that are non-empty in the column State_Name and add a column named State_Code containing the state codes of each state in column State_Name. We then added the descriptions corresponding to the entries in the Code column to prepare for the visualizations, because showing the actual descriptions in place of the code numbers will make our visualizations more easily readable. Using the processed data, we plotted the choropleth maps visualizing the costs per Medicare Enrollee in each state for the year 2020. A slider is provided for viewing the plots for different types of care, including personal health care, hospital care, etc. Since animation cannot be properly displayed here, we include screenshots of the subplots here.


Medicare Costs per Enrollee in 2020 by States and Type of Care
Y2020
4000
3500
3000
2500
2000
1500
▶ ■
Type of Care: Physician & Clinical Services
Personal Health Care
Home Health Care


Medicare Costs per Enrollee in 2020 by States and Type of Care
Y2020
6500
6000
5500
5000
4500
▶ ■
Type of Care: Hospital Care
Personal Health Care
Home Health Care


Medicare Costs per Enrollee in 2020 by States and Type of Care
Y2020
13k
12k
11k
10k
9k
▶ ■
Type of Care: Personal Health Care
Personal Health Care
Home Health Care



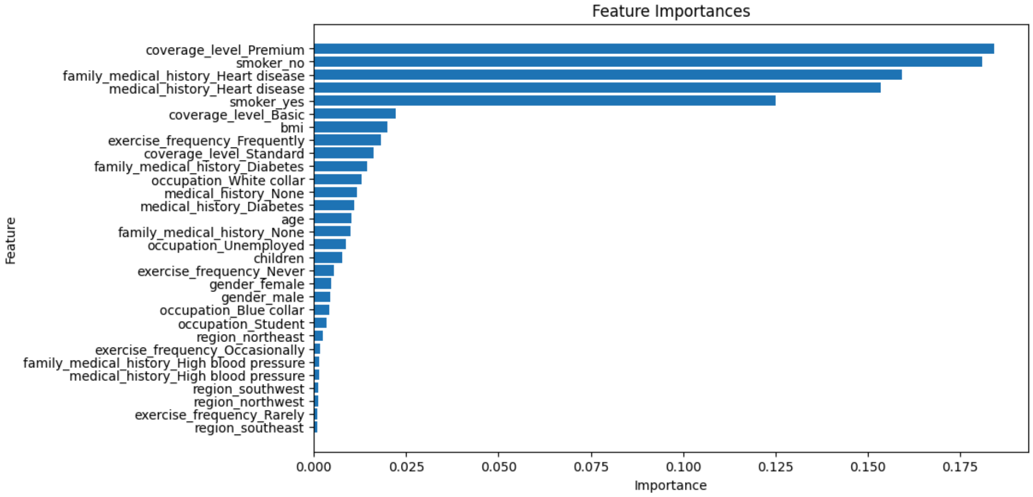
From these choropleth maps, we can observe that there exists a difference between health expenditures per capita in US states for the year 2020. Specifically, the southern, coastal states tend to have higher costs per enrollee compared to the northern, central states. In general, this trend is commonly observed for all types of care, which shows that there is a correlation between medical costs and region. This finding challenges our previous conclusion for the primary dataset that region does not have a significant effect on medical charges. Therefore, our primary dataset may not be an ideal synthesis of the real world in that it has limitations if seen as an accurate representation of real-world scenarios. This is an important observation that cannot be overlooked, and we will take this into consideration in our evaluation and reflection stage.

4. Feature Selection

4.1 Preview of Feature Importance by RF Regression

After data preparation, we have a total of 30 features with varying importance. Now, we want to investigate the effect of the number of significant features used in modeling on model accuracy and extract an appropriate number of most important features to be used in our modeling stage. In theory, this would be more computationally efficient, and focusing on a smaller portion of the features will make it easier to analyze the implications.

By fitting a random forest regressor, we found that some of the features are significantly more important than the rest. We can also tell that the middle ones are somewhat important, while the rest have negligible importance. However, it might not be a wise choice to consider just the several top features shown in the plot and remove others. We want to more explicitly determine the number of features needed for best modeling.



4.2 Number of Features by Lasso Coefficients

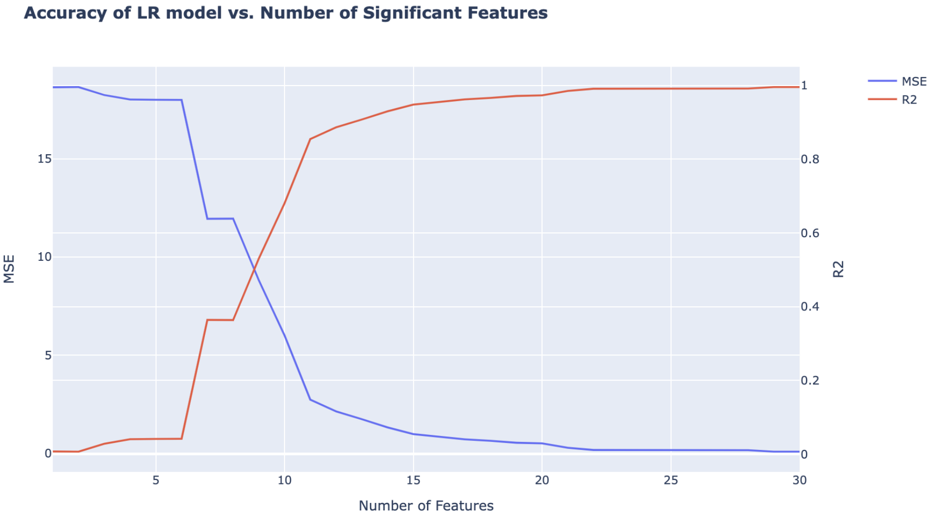
To solve this problem mentioned previously, we used lasso coefficients to determine the number of features needed. Lasso coefficient refers to the coefficients of the features in a linear regression model that has been regularized using the Lasso (Least Absolute Shrinkage and Selection Operator) method. Lasso is a type of linear regression that includes a penalty equal to the absolute value of the magnitude of the coefficients. This penalty term helps in both regularization (preventing overfitting) and feature selection.

The good thing about this process is that lasso sets the weights of non-important features to 0, so we could obtain a cutoff value for the appropriate number of features by searching for non-zero coefficients. The implementation of the Lasso model suggested to us that it is best to take the top 21 most significant features. So we redefined the scaled train and test sets by keeping only the identified most significant features.

4.3 Effect of Number of Significant Features on Accuracy by RFE

Recursive Feature Elimination (RFE) is a feature selection technique used in machine learning to select a subset of the most relevant features for model construction. The main idea behind RFE is to recursively remove the least important features from the dataset and build a model with the remaining features. The goal is to enhance the model's performance by reducing overfitting and improving generalization.

To verify our choice of significant features and to further explore the relation between number of significant features on the model accuracy, we wrote functions to apply RFE that returns MSE and R2 scores of models using different numbers of significant features. We then plotted the MSE and R2 scores vs. number of significant features together.



The plot sufficiently demonstrates that accuracy increases as we increase the number of significant features used in the model. However, the rate of increase is not constant, and there exists a threshold value beyond which the rate of increases becomes very low (almost 0). We can tell from the plot that this threshold value is somewhat around 20. Therefore, taking the 21 most significant features is an appropriate choice. We can now use this result in the modeling process.

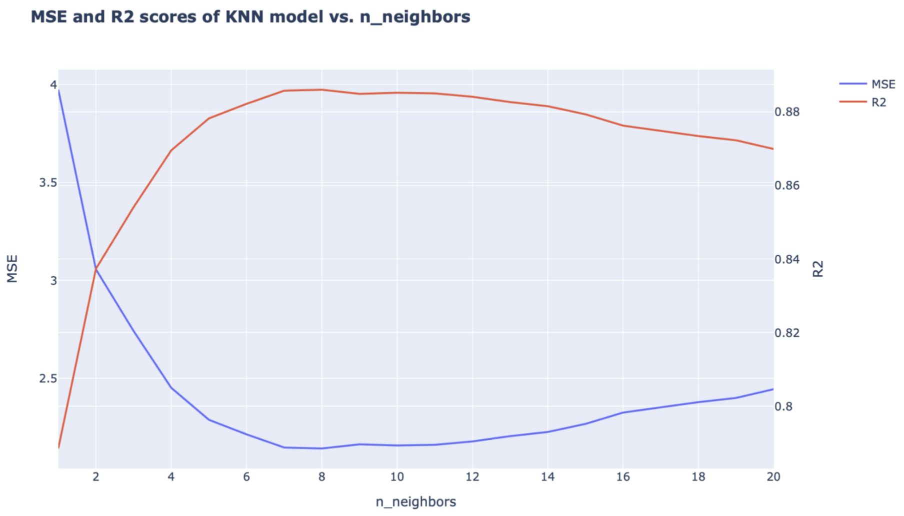
5. Modeling

5.1 Selecting Evaluation Metrics

- **MSE**: Mean Squared Error was used as the first metric to compare performance between models. This is because it measures the average of the squares of the errors, providing a clear indication of the model's prediction accuracy. In the context of the insurance project, the target variable 'charges' represents the insurance premiums. By scaling 'charges' down by a factor of 1000, we ensured that the MSE values remain manageable and easier to interpret. The MSE penalizes larger errors more than smaller ones, making it a sensitive metric for evaluating model performance. In our project, where predicting insurance premiums accurately is crucial, using MSE helps in identifying models that consistently provide close-to-actual predictions.
- **R-squared**: The R-squared (coefficient of determination) metric was used to assess the proportion of variance in the target variable ('charges') that can be explained by the features in the model. This metric provides insight into the overall fit of the model. In the context of the insurance project, the target variable 'charges' represents the insurance premiums, and R-squared indicates how well the model captures the relationship between the input features (such as age, BMI, smoking status, etc.) and the scaled insurance premiums. Since we divided 'charges' by 1000 to scale the data, the R-squared value remains unaffected by this scaling. It remains a valuable metric for understanding the model's explanatory power. A higher R-squared value indicates that the model explains a greater portion of the variance in the insurance premiums, implying that it captures the underlying patterns in the data effectively. By using R-squared, we can determine how well the features in the dataset contribute to predicting insurance premiums and compare the explanatory power of different models. This helps in selecting the best model that not only minimizes prediction errors (as indicated by MSE) but also provides a strong explanatory relationship between the input features and the target variable.

5.2 K-Nearest Neighbors

The K-Nearest Neighbor model was the first one we tested. This c algorithm makes predictions based on how similar data points (neighbors) are to the data point being predicted. To determine similarity between data points, k-NN typically uses distance metrics such as Euclidean distance. Once distances are computed, the algorithm identifies the k-nearest neighbors to the new data point. For regression tasks like predicting insurance premiums, the algorithm typically computes the average (or weighted average) of the target values of these k neighbors as the predicted value. The *n_neighbors* parameter is crucial and needs to be tuned: too small a value of k might lead to overfitting (high variance), whereas too large a value might lead to underfitting (high bias).



Plot showing MSE and R2 scores of KNN model with varying number of n_neighbors

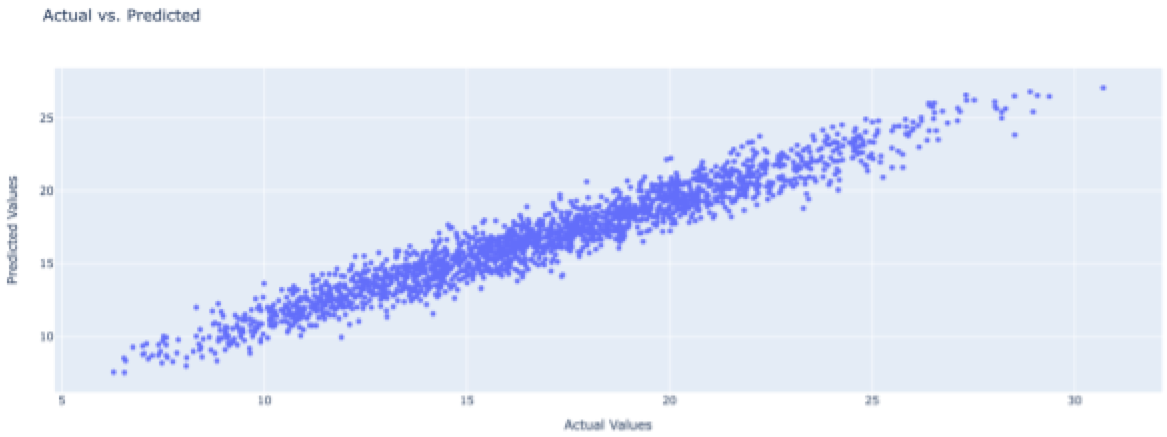
This visualization allows a comparison of both metrics on the same plot effectively. Overall, this plot helps compare how MSE and R2 scores change with different values of n_neighbors in the k-NN regressor. From the plot, it looks as though at a value of 11 n_neighbors is where the MSE and R2 values start to plateau.

In selecting the number of neighbors for the k-NN regression model, we utilized a grid search approach. This technique systematically tests different values to find the optimal combination that minimizes the MSE.

During the grid search, the model was evaluated using 5-fold cross-validation, which ensures robustness by splitting the dataset into 5 subsets and training the model on 4 subsets while validating on the remaining subset. This process helps to prevent overfitting and provides a more accurate assessment of how the model will generalize to new data.

The grid search also considered different weighting schemes (uniform and distance-based) and distance metrics (Euclidean and Manhattan) to further refine the model's performance. The combination of number of neighbors, weighting, and distance metric that resulted in the lowest MSE on the validation sets was selected as the best configuration for the k-NN model.

Best parameters: Metric: *manhattan*, n_neighbors: *10*, weights: *distance*

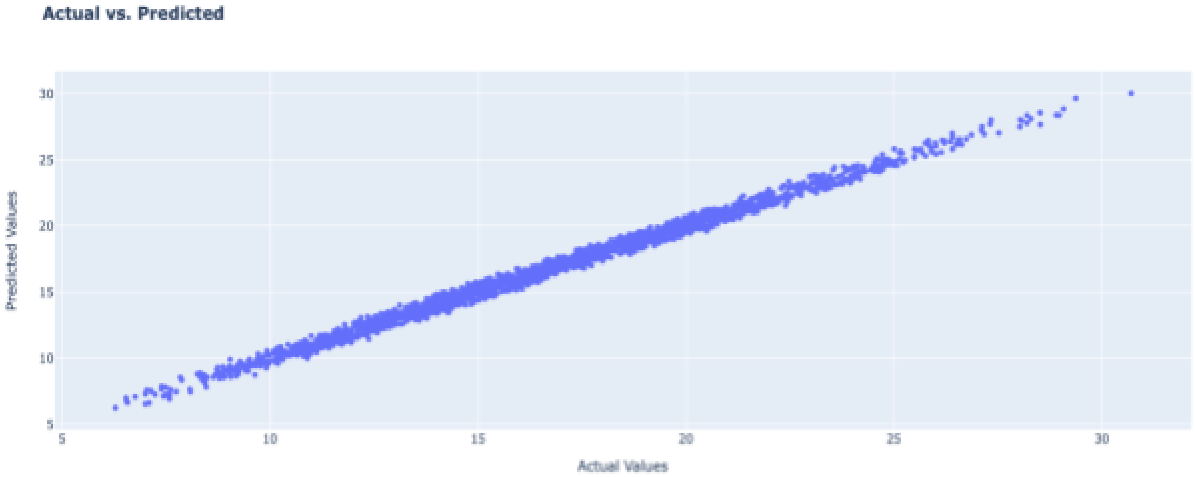


Plot showing the relationship between actual values and predicted values with the best K-NN model

MSE: 1.6
R2: 0.91

5.3 Linear Regression

Linear regression was utilized to predict health insurance premiums based on the significant features. This model aims to establish a linear relationship between the input features and the insurance premiums.



Plot showing the relationship between actual values and predicted values with the linear regression model

MSE: 0.13
R2: 0.99

5.4 Support Vector Regression (SVR)

Support Vector Regression (SVR) was implemented to predict health insurance premiums using various kernels such as RBF, polynomial, and sigmoid. The SVR model aims to find the optimal hyperplane that minimizes the prediction error while maximizing the margin between the predicted and actual values.

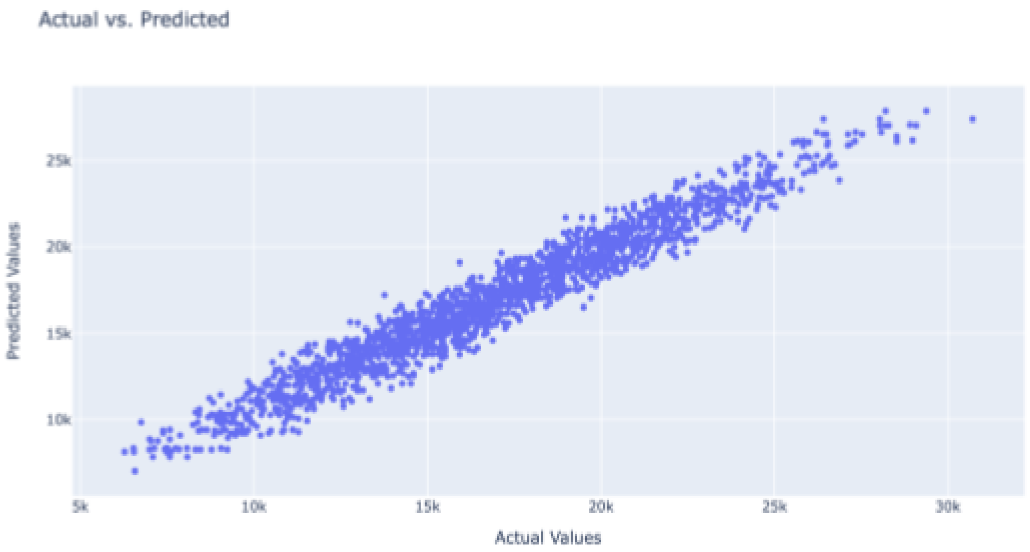
To identify the best-performing SVR model, we performed a randomized search with cross-validation to tune the hyperparameters for each kernel. The hyperparameters tuned included the regularization parameter C, the kernel coefficient γ, and the polynomial degree (for the polynomial kernel).

Randomized search was chosen over grid search for hyperparameter tuning due to time efficiency. This is because randomized search samples a subset of the hyperparameter space, making it significantly faster than grid search, which exhaustively evaluates all possible combinations.

By comparing the results from different kernels, we aimed to determine the best SVR configuration for predicting insurance premiums. Despite the flexibility and power of SVR, it was noted that the model's performance did not surpass that of the linear regression model.

Kernel	MSE	R2
rbf	0.17	0.99
sigmoid	0.23	0.98
polynomial	0.15	0.99

Polynomial kernel performed best
Best parameters:, gamma: *auto*, degree: *3*, C: *10*



Plot showing the relationship between actual values and predicted values with the best SVR model

MSE: 0.15
R2: 0.99

5.5 Neural Network

We constructed a neural network using TensorFlow's Keras API with the following architecture: two hidden layers, each comprising 64 neurons activated by ReLU functions. ReLU introduces non-linearity essential for learning complex patterns efficiently. The output layer consists of a single neuron for regression tasks, ensuring the model predicts continuous values, which is what we want as *charges* is continuous.

To optimize the model, we used the Adam optimizer, known for its adaptive learning rate and efficient convergence with ReLU activations. This choice aims to minimize the mean squared error loss function, suitable for regression problems like ours, where we aim to predict continuous insurance claim amounts.

Evaluating the model on a validation split of 10% of the training data, we aimed to prevent overfitting and ensure generalizability to unseen data.

Adjustments were made iteratively based on empirical results to strike the right balance between model complexity and generalization ability. More specifically, these were the hyperparameters we fine tuned:

- **Batch Size:** Initially, we focused solely on batch size, keeping the number of neurons per layer at 64, the number of hidden layers at 2, and the number of epochs at 20.

1. We began with a batch size of 64 and reduced it incrementally as long as the Mean Squared Error (MSE) decreased.
2. If MSE increased after reaching a certain batch size, we reverted to the previous batch size and then reduced the batch size in smaller increments.

By methodically adjusting the batch size and observing its impact on model performance, we were able to identify the optimal batch size that minimized MSE, ensuring the model balanced accuracy and generalization.

Batch Size	Test Loss (MSE)
64	0.256
45	0.238
25	0.220
15	0.224
20	0.253

From the table, we can observe that the test loss (MSE) decreased as the batch size was reduced from 64 to 25. However, when the batch size was further reduced to 15, the MSE slightly increased. This suggests that the optimal batch size for our model, based on the given settings, is 25, as it provided the lowest test loss (MSE).

- **Hidden Layer Number:** Number of Hidden Layers: After identifying the optimal batch size, we kept the batch size fixed at 25, and then varied the number of hidden layers while keeping the number of neurons per layer at 64 and the number of epochs at 20.

Hidden Layer Number	Test Loss (MSE)
2	0.220
3	0.211
4	0.214

From these results, we see that the test loss (MSE) was minimized when using 3 hidden layers. Adding more hidden layers beyond this point led to a slight increase in the test loss, indicating that the model was becoming more complex and possibly overfitting to the training data.

- **Number of Neurons per Hidden Layer:** With the optimal batch size (25) and number of hidden layers (3) established, we then varied the number of neurons in each hidden layer. We compared models with 32, 64, and 128 neurons per layer.

Number of Neurons	Test Loss (MSE)
64	0.211
32	0.226
128	0.307

  The results indicate that having 64 neurons per hidden layer resulted in the lowest test loss (MSE). Reducing the number of neurons to 32 or increasing it to 128 both led to higher MSE values. This suggests that 64 neurons per layer provided the best balance between model complexity and generalization ability.

- **Number of Epochs:** After determining the optimal batch size, number of hidden layers, and number of neurons per layer, we varied the number of epochs while keeping the other hyperparameters fixed at a batch size of 25, number of hidden layers of 3 and number of neurons per hidden layer of 64. The number of epochs controls the number of times the model iterates over the entire training dataset during training.

Number of Epochs	Test Loss (MSE)
20	0.211
15	0.232
25	0.264

  The results indicate that having 20 epochs resulted in the lowest test loss (MSE).

Therefore, putting it all together, best neural network model performance was with

- batch size = 25
- number of hidden layers = 20
- number of neurons in hidden layer = 64
- epoch number = 20

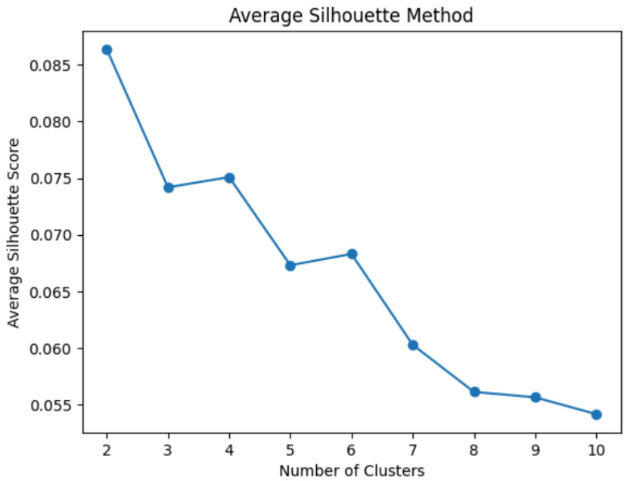
MSE: 0.211

5.6 Clustered Linear Regression

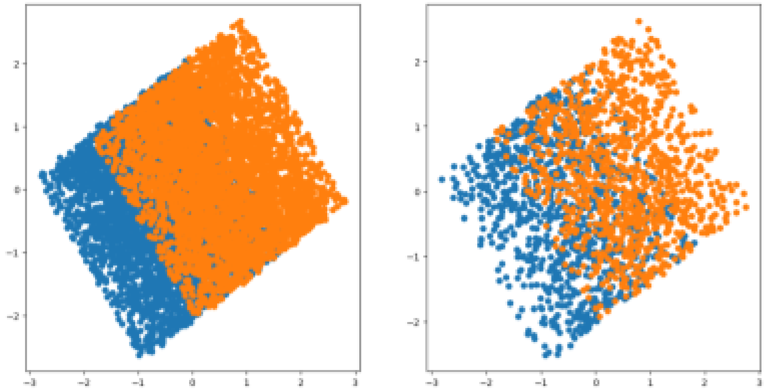
The final model we trained and tested on our data was clustered linear regression. This model is known to improve the linear regression model that was previously used and determine if our data experiences any clustering. If clustering is found, we might be able to more accurately predict exactly what features are contributing to individual's health insurance coverage levels.

To determine the best number of clusters for our model, we used the average silhouette method to determine the optimal number of clusters. With this method, the optimal number of clusters is determined by the maximum point on the graph or the highest average silhouette score. With our data, and as we can tell in our graph below, the optimal number of clusters for our model is 2 so we used two clusters for our Clustered Linear Regression model.

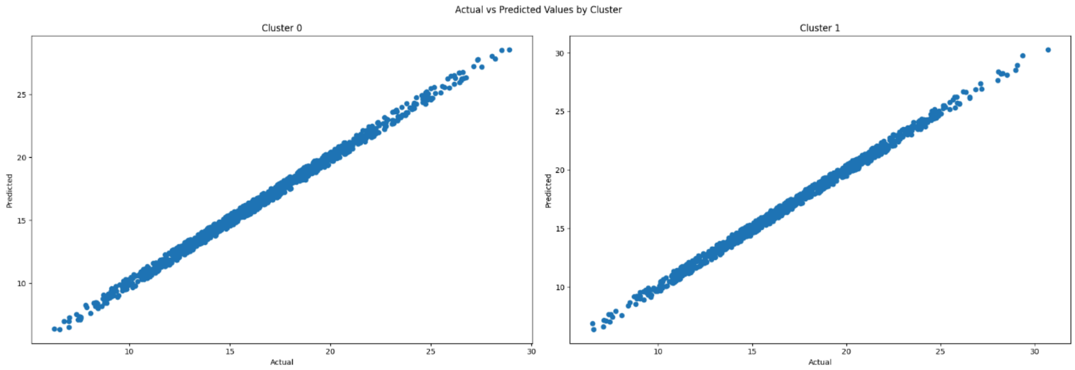
```
Best number of clusters based on silhouette score: 2
```



Using this determined optimal number of clusters, we built and trained our model. We found our model did exceptionally well.



Visualization of Clusters in 2-D



Cluster 0:
MSE: 0.089
R2: 0.995
Cluster 1:
MSE: 0.083
R2: 0.996

Before, our best model as reflected by the scores was linear regression but in comparison, this new model scored better. This also makes sense because clustered linear regression is supposed to improve the accuracy of normal linear regression which is exactly what was reflected in our CLR model.

After further analysis by examining cluster centroids, we found that we could possibly predict what features are contributing to our clusters:

- Cluster 0: This cluster might represent younger individuals with slightly lower BMI, fewer children, a higher proportion of smokers, more people who never or occasionally exercise, slightly more blue-collar workers and unemployed individuals, and a higher proportion of people with Premium coverage.
- Cluster 1: This cluster might represent older individuals with slightly higher BMI, more children, a lower proportion of smokers, more people who exercise rarely, slightly more students and white-collar workers, and a higher proportion of people with Standard coverage.

By analyzing these centroids, we can infer that Cluster 0 tends to consist of younger, possibly single individuals or couples without children, with slightly higher smoking rates and are receiving higher insurance coverage. Cluster 1 tends to consist of older individuals, possibly families with children, with slightly lower smoking rates, and are receiving Standard coverage.

These interpretations are useful when trying to predict health premiums and insurance charges.

6. Improvement and Extension

6.1 PCA Dimension Reduction

To further explore all the models, and see whether the performance could be improved, we attempted to perform PCA dimension reduction on our encoded dataset (instead of Lasso) and see if that yielded better results. After standardization, the explained variance ratio remained relatively low despite modifying the *n_components* number:

N_components in PCA	Explained Variance (%)
2	24
5	48
10	65

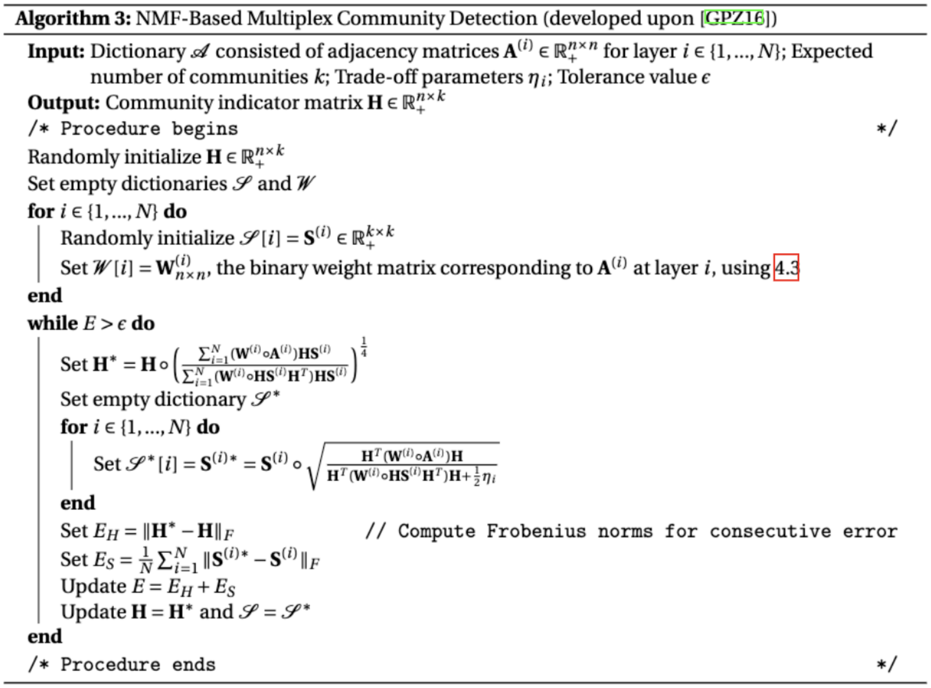
Despite the relatively low explained variance, we tested the performance of PCA with 10 components in the linear regression model and compared it to the performance of the linear regression model with the original feature selection technique. Despite this, performance was worse, suggesting that PCA, despite reducing dimensionality, significantly worsened performance with our dataset.

MSE: 4.87
R2: 0.74

6.2 Network Approaches

To look into more possible approaches and interpretations we can apply and derive from this dataset, we experimented on some network analysis methods learned from other courses. We were especially interested in analyzing the clustering pattern of this dataset, since a clustered pattern showed up in the CLR model implemented in the previous section. In many studies, clustering individuals to subpopulations based on individual attributes (e.g genetic data) has become commonplace. With the knowledge that individuals can be connected in a network by assigning weighted edges between them representing how similar their attributes are, we applied a multilayer community detection method which has the potential to retain more of the data's structural information. However, since this dataset is not specifically designed for network analysis purposes, the result may be not very informative. Despite this, we hope this analysis could shed light on how data structures similar to our dataset can be studied using these methods.

The algorithm, named Weighted Simultaneous Symmetric Non-Negative Matrix Tri-Factorization (WSSNMTF), is proposed in this paper. This community detection method detects clustering of multi-layer graphs and is robust with respect to missing edges and noise. It is developed based on an NMF framework, which is a classic topic modeling technique used in machine learning. The algorithm is explained in detail in this paper, but code is not provided. Therefore, we defined the functions needed for implementing this algorithm based on the descriptions in paper, and wrote an outline for the algorithm using latex:



Algorithm 3: NMF-Based Multiplex Community Detection (developed upon [GPZ16])

```
Input: Dictionary $\mathscr{A}$ consisted of adjacency matrices $\mathbf{A}^{(i)} \in \mathbb{R}_{+}^{n \times n}$ for layer $i \in \{1, ..., N\}$; Expected
        number of communities $k$; Trade-off parameters $\eta_i$; Tolerance value $\epsilon$
Output: Community indicator matrix $\mathbf{H} \in \mathbb{R}_{+}^{n \times k}$
/* Procedure begins                                                              */
Randomly initialize $\mathbf{H} \in \mathbb{R}_{+}^{n \times k}$
Set empty dictionaries $\mathscr{S}$ and $\mathscr{W}$
for $i \in \{1, ..., N\}$ do
    Randomly initialize $\mathscr{S}[i] = \mathbf{S}^{(i)} \in \mathbb{R}_{+}^{k \times k}$
    Set $\mathscr{W}[i] = \mathbf{W}_{n \times n}^{(i)}$, the binary weight matrix corresponding to $\mathbf{A}^{(i)}$ at layer $i$, using 4.3
end
while $E > \epsilon$ do
    Set $\mathbf{H}^{*} = \mathbf{H} \circ \left( \frac{\sum_{i=1}^{N} (\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)}) \mathbf{H} \mathbf{S}^{(i)}}{\sum_{i=1}^{N} (\mathbf{W}^{(i)} \circ \mathbf{H} \mathbf{S}^{(i)} \mathbf{H}^{T}) \mathbf{H} \mathbf{S}^{(i)}} \right)^{\frac{1}{4}}$
    Set empty dictionary $\mathscr{S}^{*}$
    for $i \in \{1, ..., N\}$ do
        Set $\mathscr{S}^{*}[i] = \mathbf{S}^{(i)*} = \mathbf{S}^{(i)} \circ \sqrt{\frac{\mathbf{H}^{T} (\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)}) \mathbf{H}}{\mathbf{H}^{T} (\mathbf{W}^{(i)} \circ \mathbf{H} \mathbf{S}^{(i)} \mathbf{H}^{T}) \mathbf{H} + \frac{1}{2} \eta_i}}$
    end
    Set $E_H = \|\mathbf{H}^{*} - \mathbf{H}\|_F$          // Compute Frobenius norms for consecutive error
    Set $E_S = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{S}^{(i)*} - \mathbf{S}^{(i)}\|_F$
    Update $E = E_H + E_S$
    Update $\mathbf{H} = \mathbf{H}^{*}$ and $\mathscr{S} = \mathscr{S}^{*}$
end
/* Procedure ends                                                                */
```

To apply this algorithm, we need to construct a network. We treat each individual as a node in the network, and the health data taken from each individual becomes a unique data vector for this node. Here we chose to use cosine similarity to assign weight for each pair of nodes. We then approached the problem by experimenting on a 2-layer structure. This algorithm has high time complexity, so we took a subset of 500 nodes to reduce the computation power needed. We then performed label encoding on the original dataset to convert the categorical features to numerical, so that data of each individual can be represented by a vector. We then standardized the features in the same way as we did before.

We selected 7 representative features to form the 2-layer structure. Edges in the first layer represent the similarity between individuals based on general attributes (age, bmi, children, and gender). Edges in the second layer represent the similarity between individuals based on medical profile (coverage level, medical history, and family medical_history). This way, we generated 2 adjacency matrices to be input in the algorithm as a 2-layer structure.

We ran 3 experiments, trying to detect 2, 3, or 4 communities in our data and compare the outputs. We assessed the clustering results by using modularity. Modularity is a measure used in network theory to quantify the strength of division of a network into communities. A higher modularity value indicates a stronger community structure, where nodes within the same community are more densely connected to each other than to nodes in different communities.

The 3 experiments showed that we got low modularity scores, implying a weak clustering pattern in our dataset. Despite these low scores, this outcome is not without value. The results highlight important aspects of our dataset and suggest areas for future research. Specifically, the weak clustering pattern may indicate that our current choice of layered structures and selected features do not capture the intrinsic community structures well, suggesting a need for more sophisticated feature engineering or the inclusion of additional data sources.

This experience has been invaluable in enhancing our understanding of network analysis and modularity. It has underscored the complexity of community detection in real-world datasets and the challenges inherent in accurately modeling such structures. Furthermore, these findings pave the way for several potential future studies. We could explore advanced community detection algorithms, such as the Louvain or Leiden methods, which may provide more nuanced insights into the community structure. Another avenue for improvement is experimenting with different similarity metrics or incorporating node and edge attributes that might better capture the relationships within the data.

7. Evaluation and Conclusion

7.1 Advantages and Limitations

Our project analyzes the various factors correlated with health care plans and explores a variety of models in predicting health premiums. Our models achieved a high accuracy overall and provided us with meaningful insights on the topic. However, there are limitations in our approaches:

- Due to the synthetic nature of this insurance dataset, this data cannot be treated as an ideal, accurate representation of the real world scenario. Specifically, BMI is not normally distributed in our dataset as it is in the real world.
- There is disparity between factors we detected to be important and factors actually affecting health premiums. For example, according to HealthCare.gov, location should be a major determinant affecting health premiums (and we did observe this pattern in supplementary data analysis of the national health expenditure data), but we did not detect a strong correlation in our dataset.
- We based our work only on a small subset due to limiting computation power, so the scale of analysis might not be sufficiently extensive and comprehensive.

In future opportunities, we expect to dive more into this topic and overcome these challenges. A more detailed approach of studying such datasets with network methods can be developed to improve the clustering accuracy of community detection. We can also incorporate data from other reliable sources and experiment with different numbers of features. And we can test other dimension reduction methods like kernel PCA.

7.2 Conclusion

In conclusion, our project provided a detailed analysis of the factors correlated with health care plans and explored various models for predicting health premiums. We performed data analysis and visualization, compared performance of multiple models using MSE and R2, and established a best performing model using the clustered linear regression. While studying the useful implications, we encountered limitations due to multiple reasons listed before, which directed us onto the path for more improvements. These future endeavors will enhance the robustness and applicability of our findings in real-world health care plan analysis.

8. Group Member Roles

Anshika Khandelwal

1. Visualization of relationships between categorical variables vs charges
2. The modeling section: train all models (KNN, linear regression, SVR, neural network)
3. PCA Dimension Reduction

Dakota Lin

1. Visualization of relationships features and target (charges vs. BMI for different gender and smoking status & charges vs. age for different medical history)
2. Supplementary data analysis: national health expenditure data by CMS
3. Data preparation for training (feature encoding, standardization, train and test sets)
4. Feature selection using RFE and Lasso and effect of number of significant features on accuracy
5. Network approaches for community detection

Sarah Ward

1. Visualization of relationships between numerical variables vs coverage level
2. Clustered Linear Regression Model: train, test, visualize model and predict its performability including centroid clusters
3. Detailed descriptions of dataset features in introduction, visualizations, and CLR

9. References

- **Primary dataset**: “Insurance Dataset for Predicting Health Insurance Premiums in the US." https://www.kaggle.com/datasets/sridharstreaks/insurance-data-for-machine-learning/data
- **Supplementary dataset**: “National health expenditure data.” Centers for Medicare & Medicaid Services https://www.cms.gov/data-research/statistics-trends-and-reports/national-health-expenditure-data/state-residence
- **WSSNMTF algorithm**: V. Gligorijević, Y. Panagakis and S. Zafeiriou, "Fusion and community detection in multi-layer graphs," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016, pp. 1327-1332, doi: 10.1109/ICPR.2016.7899821.
- US Government directions on how insurance companies set health premiums: https://www.healthcare.gov/how-plans-set-your-premiums/#:~:text=Five%20factors%20can%20affect%20a,can't%20affect%20your%20premium.