Report for Personal Project

Hiep (Andrew) Nguyen

I. Abstract:

Predicting the future healthcare costs of individuals is the hardest step that insurance companies need to go through before offering the premiums level their customers have to pay regularly. Recently, actuaries have begun to apply ML models to this task to save time and maximize accuracy as well. In this project, I compiled, analyzed, evaluated, and then compared performances of eight potential regression models for predicting healthcare costs. Using the health data of almost 1000 customers as the dataset for this comparative analysis, I found that random forest and gradient boosting had the best predictive performances. I also tried to build an appropriate regression neural network which can apply to much more complicated datasets.

II. Introduction:

Insurance is a tool that individuals and businesses use to protect themselves from detrimental financial consequences. In most cases, people or businesses that have insurance pay premiums regularly. Because this problem relates directly to companies' revenue and long-term growth, predicting accurate healthcare costs for individuals is one of the most important tasks for the operation of a health insurance company.

This task is also important for various stakeholders beyond health insurers. It helps the governments gain the necessary insights to control insurance companies and prevent them from manipulating the market. Moreover, for the insured (the person covered under the insurance policy), knowing in advance their likely expenditures for the next year could potentially help them to choose appropriate insurance plans.

In essence, health insurance companies price these the premium based on the probability of certain events (risks) occurring among a pool of people [1]. In particular, actuaries who are in charge of this work essentially use probability theorems (e.g. Law of large numbers) and statistical models to quantify the risk level. However, insurers still face an inevitable issue - imperfect information. Indeed, we cannot predict future events with 100% certainty, and it makes estimating the future risk extremely difficult.

Fortunately, besides developing sophisticated mathematical models "organically", with the development of data science, actuaries today can utilize datasets about their customers' characteristics to develop many different ML models to increase the accuracy of risk predictions. Moreover, there are some studies have shown that gradient boosting and ridge regression have good performances in predicting medical risk [2]. For this reason, I want to make a systematic comparison of supervised learning models for predicting the future medical expenses of individuals. I also built an appropriate regression neural network which can be used for datasets containing features and targets that have complex non-linear relationships.

III. Background

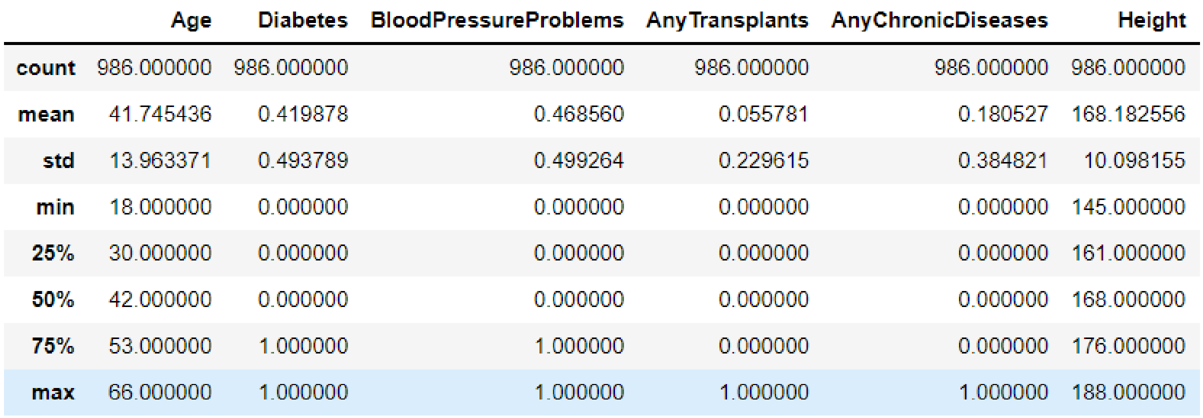
Using machine learning approaches to predict medical costs of patients is not a new topic. There have been several scientific research chosen different regression models to predict healthcare costs in general, or even in some specific branches like predicting high-cost high need patient expenditures (used ordinary least squares linear regression, LASSO, gradient boosting machine, and recurrent neural networks) [3], healthcare cost of breast cancer patients (used DBSCAN and Markov chain algorithm) [4], … Like house prices or stock cost, there is no common framework for a best model, and researchers still create and compare new models to have better performances. I will make a comparison of eight models including linear regression, ridge regression, LASSO, XGB regressor, gradient boosting regressor, decision tree, random forest, and neural network.

IV. Data

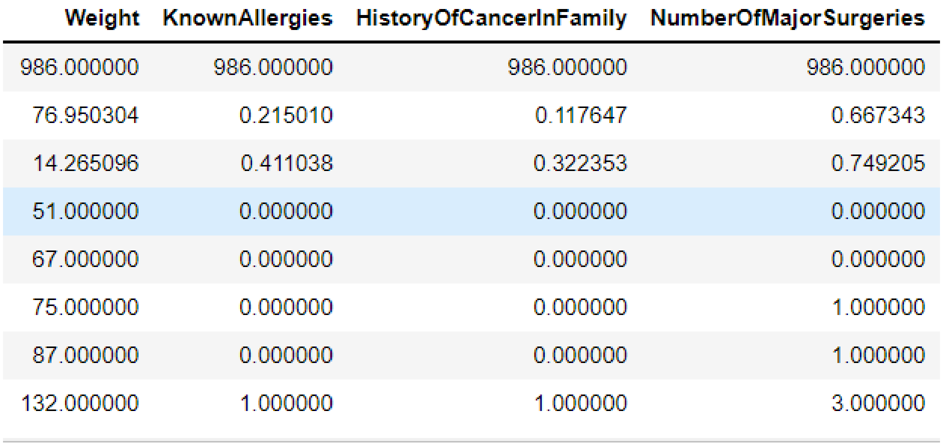
I used the dataset released by a medical insurance company, and found it on Kaggle; the dataset records medical information of almost 1000 customers and their yearly premium. There are 10 features:

1. Age (numerical): age of customer
2. Diabetes (binary): whether the person has abnormal bloodSugar levels
3. BloodPressureProblems (binary): whether the person has abnormal blood pressure levels
4. AnyTransplants (binary): any major organ transplants
5. AnyChronicDiseases (binary): whether customer suffers from chronic ailments like Asthama, etc.
6. Height (numerical): height of customer
7. Weight (numerical): weight of customer
8. KnownAllergies (binary): whether the customer has any known allergies
9. HistoryOfCancerInFamily (binary): whether any blood relative of the customer has had any form of cancer
10. NumberOfMajorSurgeries (numerical): the number of major surgeries that the person has had

Here is the descriptive statistics information of variables:



	Age	Diabetes	BloodPressureProblems	AnyTransplants	AnyChronicDiseases	Height
count	986.000000	986.000000	986.000000	986.000000	986.000000	986.000000
mean	41.745436	0.419878	0.468560	0.055781	0.180527	168.182556
std	13.963371	0.493789	0.499264	0.229615	0.384821	10.098155
min	18.000000	0.000000	0.000000	0.000000	0.000000	145.000000
25%	30.000000	0.000000	0.000000	0.000000	0.000000	161.000000
50%	42.000000	0.000000	0.000000	0.000000	0.000000	168.000000
75%	53.000000	1.000000	1.000000	0.000000	0.000000	176.000000
max	66.000000	1.000000	1.000000	1.000000	1.000000	188.000000



Weight	KnownAllergies	HistoryOfCancerInFamily	NumberOfMajorSurgeries
986.000000	986.000000	986.000000	986.000000
76.950304	0.215010	0.117647	0.667343
14.265096	0.411038	0.322353	0.749205
51.000000	0.000000	0.000000	0.000000
67.000000	0.000000	0.000000	0.000000
75.000000	0.000000	0.000000	1.000000
87.000000	0.000000	0.000000	1.000000
132.000000	1.000000	1.000000	3.000000

Figure 1,2: Descriptive statistics information of variables

While exploring this dataset, I found that two variables 'Height' and 'Weight' individually have no means to express the health conditions of someone. Instead, we can consider BMI (numerical) which is a measure of body fat based on height and weight. It is much more useful in this case.
Formula: BMI = weight (kg) ÷ height^2 (m)

I ran a random forest model for the original data to figure out important features. Age, BMI (as expected), and NumberOfMajorSurgeries are most important features of this model.

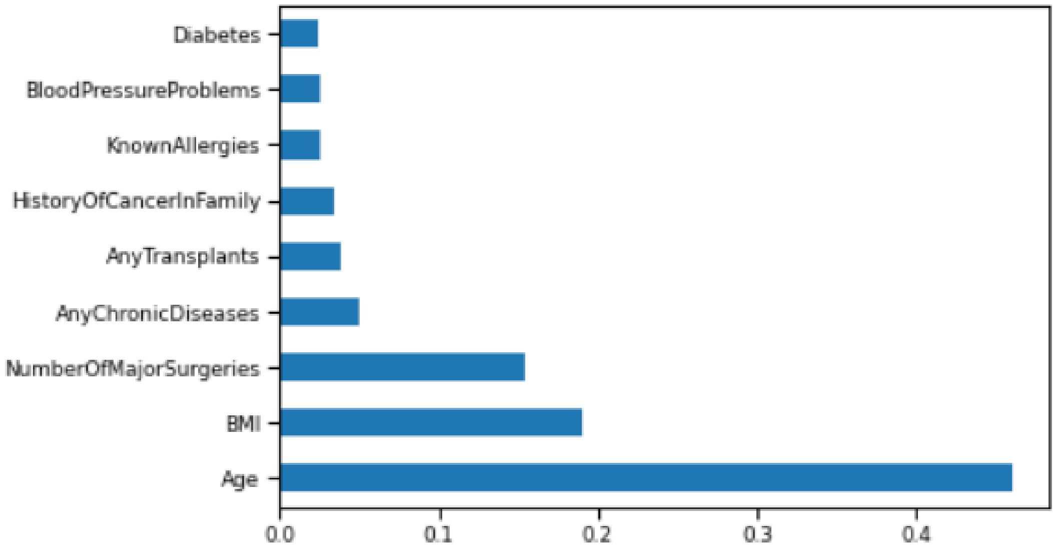
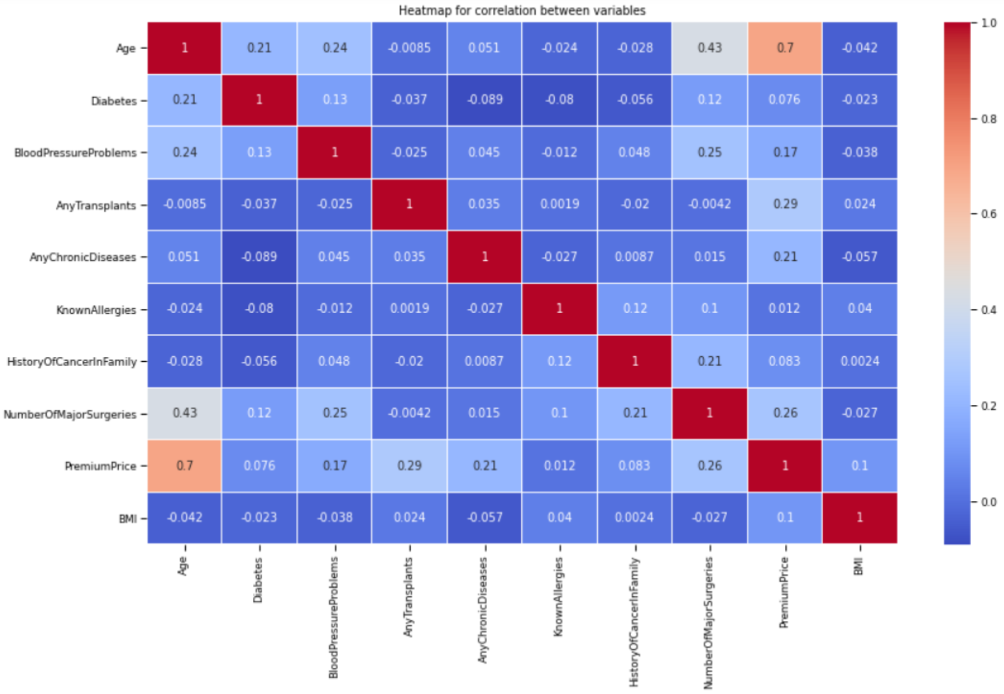


Figure 3: Important features for the random forest model

To double check, I also created the heatmap for the correlation between the 9 features:

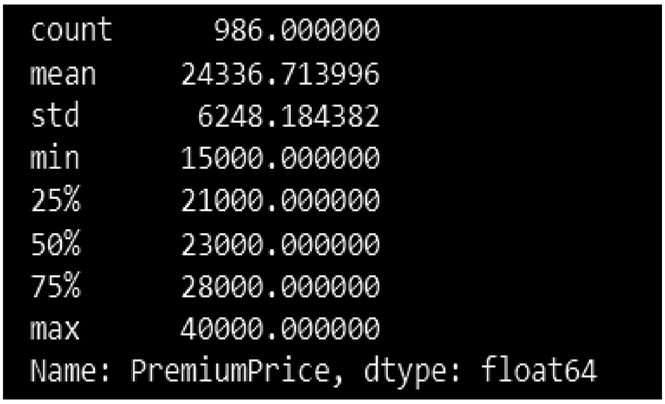


	Age	Diabetes	BloodPressureProblems	AnyTransplants	AnyChronicDiseases	KnownAllergies	HistoryOfCancerInFamily	NumberOfMajorSurgeries	PremiumPrice	BMI
Age	1	0.21	0.24	-0.0085	0.051	-0.024	-0.028	0.43	0.7	-0.042
Diabetes	0.21	1	0.13	-0.037	-0.089	-0.08	-0.056	0.12	0.076	-0.023
BloodPressureProblems	0.24	0.13	1	-0.025	0.045	-0.012	0.048	0.25	0.17	-0.038
AnyTransplants	-0.0085	-0.037	-0.025	1	0.035	0.0019	-0.02	-0.0042	0.29	0.024
AnyChronicDiseases	0.051	-0.089	0.045	0.035	1	-0.027	0.0087	0.015	0.21	-0.057
KnownAllergies	-0.024	-0.08	-0.012	0.0019	-0.027	1	0.12	0.1	0.012	0.04
HistoryOfCancerInFamily	-0.028	-0.056	0.048	-0.02	0.0087	0.12	1	0.21	0.083	0.0024
NumberOfMajorSurgeries	0.43	0.12	0.25	-0.0042	0.015	0.1	0.21	1	0.26	-0.027
PremiumPrice	0.7	0.076	0.17	0.29	0.21	0.012	0.083	0.26	1	0.1
BMI	-0.042	-0.023	-0.038	0.024	-0.057	0.04	0.0024	-0.027	0.1	1

Figure 4: Correlation map for 9 features and target.

We may consider using only Age, BMI, NumberOfMajorSurgeries , and AnyTransplants to build next regression models.

PremiumPrice is our final target. It is yearly premium of around 1000 customers



```
count       986.000000
mean      24336.713996
std        6248.184382
min       15000.000000
25%       21000.000000
50%       23000.000000
75%       28000.000000
max       40000.000000
Name: PremiumPrice, dtype: float64
```

Figure 5: Descriptive statistics information of targets

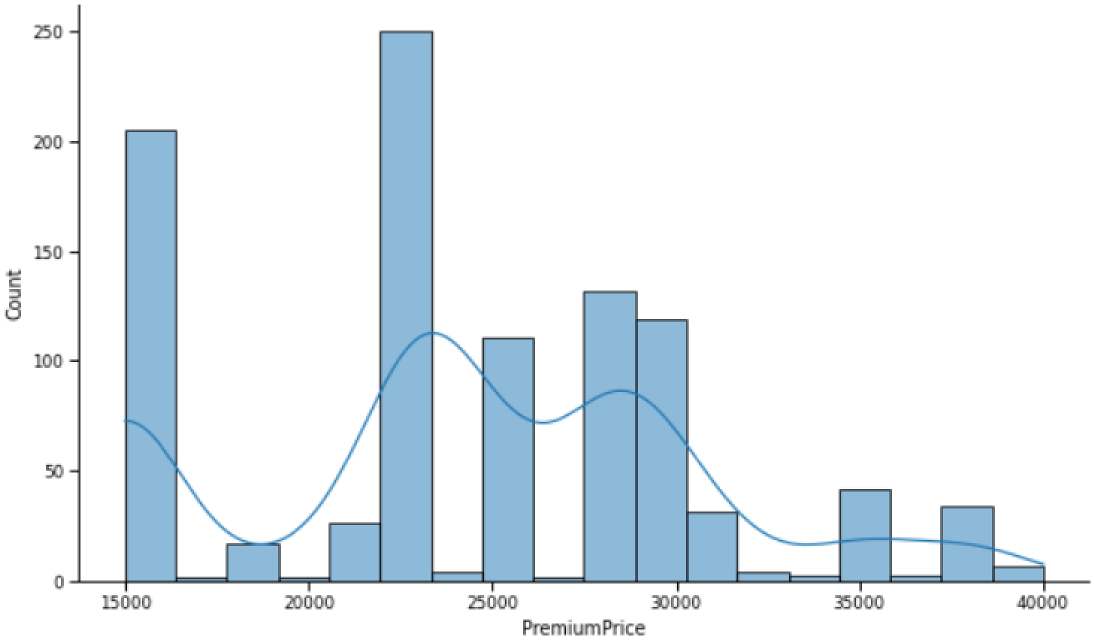
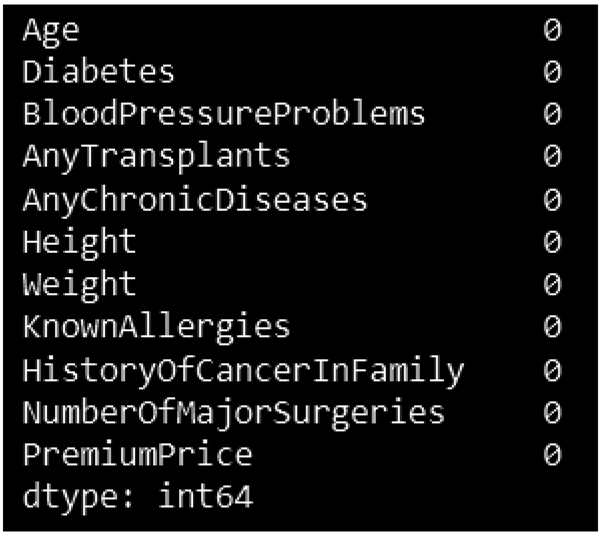


Figure 6: Distribution of targets

We can recognize that our target does not have a normal distribution and contains a number of outliers. This could make the later MSEs (mean squared errors) higher than expected.

V. Methods

The first step of my project was getting and preprocessing data. After loading data, I checked whether our dataset has missing values. It helps me to know that I did not need to use some statistical techniques such as replacing with the mean values for missing values.



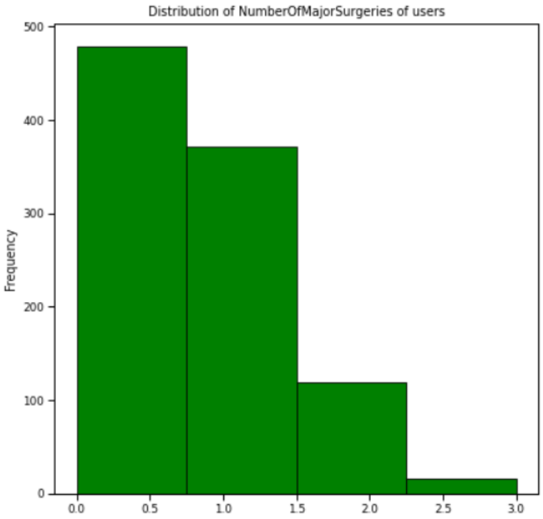
```
Age                         0
Diabetes                    0
BloodPressureProblems       0
AnyTransplants              0
AnyChronicDiseases          0
Height                      0
Weight                      0
KnownAllergies              0
HistoryOfCancerInFamily     0
NumberOfMajorSurgeries      0
PremiumPrice                0
dtype: int64
```

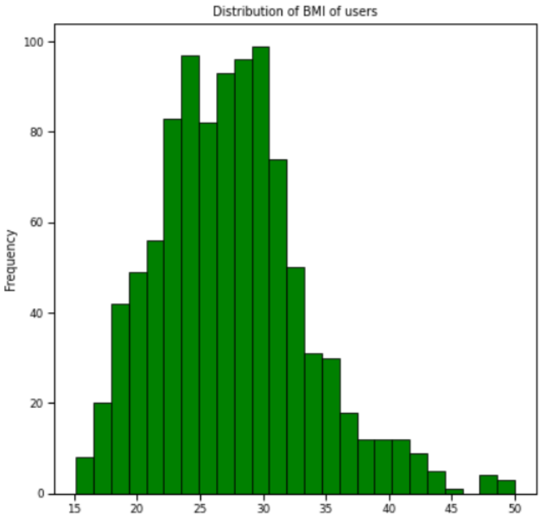
Figure 7: Checking null value of the dataset

I also recognized that all of 9 features and targets are numerical or binary which are appropriate for regression models. Thus, I did not need to use one-hot encoding which is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

The next step was exploratory data analysis (EDA). As mentioned above, we plotted the heat map for the correlation of 9 features and build a random forest model to find some important features of this dataset (plot and heatmap are in Data section). And I found that important features were Age, BMI, NumberOfMajorSurgeries , and AnyTransplants.

I then plotted the distribution of targets and important variables to detect the outliers. This could help us have initial expectations about the accuracy of models and as well as get the right insight from the data.





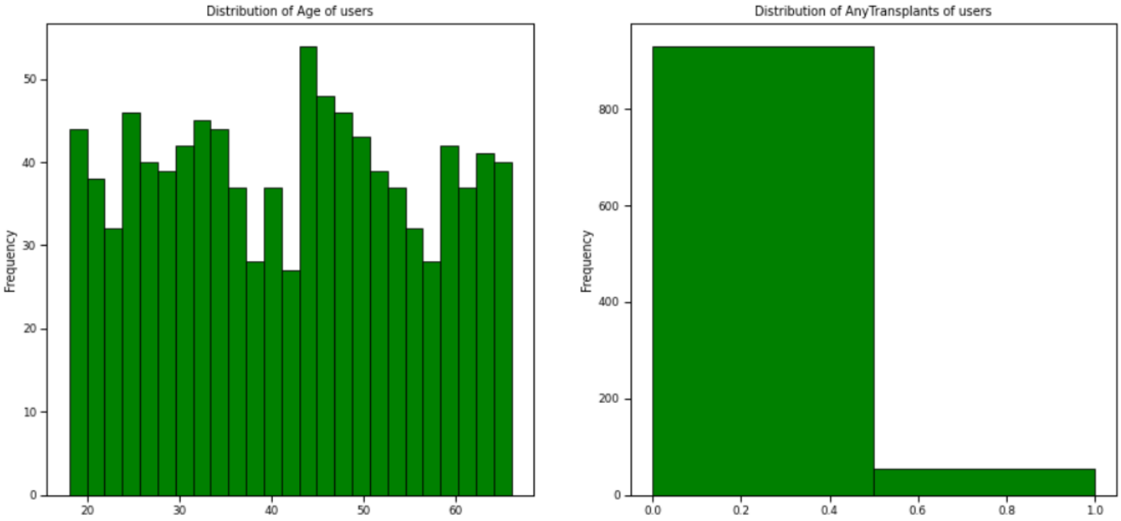


Figure 8,9: Distribution of important features

Because the target of my dataset is the yearly premium that potential patients had to pay which is continuous numerical data and our models, I definitely will choose regression models to predict it. Works on Kaggle often just focus on Linear Regression, while scientific research tried LASSO, ridge, and neural network (with a much more complicated dataset). I then decided to eight different models (from basic to advanced) to have a broader view of this problem.

Next, I split the dataset into training (70%) and testing dataset (30%). Because variables of dataset have different ranges, I used StandardScaler() to normalize data, then fitted and transformed X_train and transformed only for X_test.

To evaluate the performances of models, I used two loss functions MSE (mean squared errors), MAE (mean absolute errors), and r2_score - a very important metric that is used to evaluate the performance of a regression-based machine learning model.

Linear regression:

The first model that I used was linear regression. After fitting the training set to the the model, I used testing data to check the performance of the model, and the score was around 0.69. To evaluate the model, we use K-fold cross-validation with cv = 10 to check if we trust the model's score. The mean of the validation scores array was slightly lower than score of the testing dataset (approximate 0.58), and the standard deviation of the validation scores array is quite small(just around 0.09). Therefore, I partially trust the score 0.69 of this model.

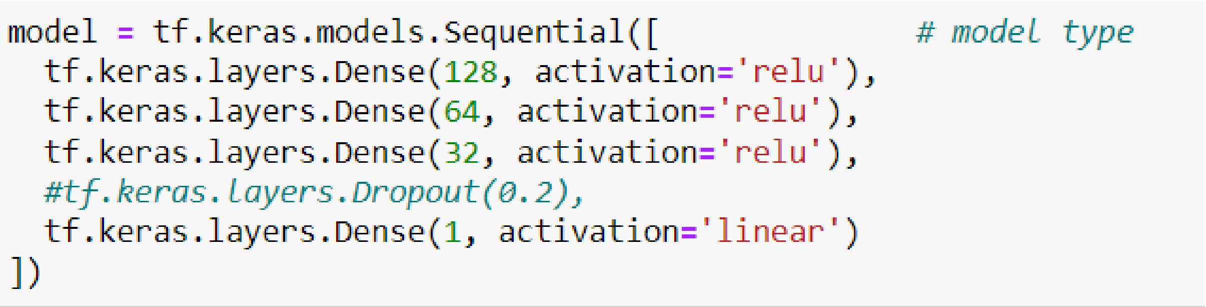
Ridge Regression , Lasso Regression:

I applied the same method to these two models. I created a loop to find the best alpha, and both models had best performance when alpha = 0.5. Actually, performances of these two model were nearly same as of linear regression.

XGB, Gradient boosting, and Random Forest:
I still applied the same method to these there models (without finding best alpha). Gradient boosting machine had best performances with the scores were around 0.8 It matches with result of previous studies.

Random Forest:
Before fiting the data into this model, I tried to use GridSearchCV to find the best value of parameters first. The result was 'max_depth' = 6, 'min_samples_leaf' = 2, 'min_samples_split ' = 2, and 'n_estimators' = 100. This models then had the best performance among all of models with the scores 0.82. It also had the lowest MSE.

Neural Network:
After EDA step, I found that all of variables of this dataset just have linear relationships. Thus, I expected the performance to be unimpressive or even worse than that of random forest or gradient boosting machine. However, I still tried to build a NN model for much more complicated datasets I may have in the future. After several attempts, I decide to use the model with this structure:



```
model = tf.keras.models.Sequential([          # model type
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(32, activation='relu'),
  #tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(1, activation='linear')
])
```

I changed the activation function of output layer to ‘linear’ to match the requirement of this problem.

I then developed a GridSearchCV model to find the best value of parameters, and the result was 'batch_size' = 16, 'epochs' = 150, 'optimizer' = 'adam'. Scores of this NN model then increased to 0.64 and MSE decreased to around 2230.

VI. Results

	Testing Score	Validation score	STD of Validation score	MSE	MAE
Linear Regression	0.69	0.58	0.09	12859369	2661

Ridge	0.69	0.58	0.09	12864249	2662
Lasso	0.7	0.59	0.09	12859267	2661
XGB	0.73	0.62	0.12	11312241	1818
Gradient boosting	0.8	0.7	0.1	8503516	1837
Decision tree	0.59	0.53	0.2	17469594	**1584**
Random Forest	**0.82**			**7791806**	1634
Neural network	0.64				2231

As we can see, the scores of most models fluctuate from 0.69 to 0.82 (except for decision tree and neural network). I also tried my best to maximum the score of NN model from intial score of 0.38. 0.64 was higher than I expected.

The purspose of my project is to create a systematic comparison of regression models for predicting medical cost. Base on results I got, I found that random forest and gradient boosting provides the best predictions. This result matchs with the finding of some previous studies. NN model is also really promising for other complicated dataset.

VII. Conclusions

The limitation my project came for the simplicity of dataset. I acknowlege that, in the real life, actuaries use datasets with around 30 to 40 variable to have the best information about their customers.

Based on our score, we conclude that random forest and gradient boosting are models that we should consider when predicting healthcare cost. For future studies , I intend to spending more time on finding a best structure for NN model. I also want to try a different dataset given by other insurance company to check whether it will give a different result.

VIII. References

[1] Steven A. Greenlaw. (2017). Principles of Economics 2e. OpenStax. https://openstax.org/books/principles-economics-2e/pages/16-2-insurance-and-imperfect-information

[2] Morid, M. A., Kawamoto, K., Ault, T., Dorius, J., & Abdelrahman, S. (2018). Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2017, 1312–1321.

[3] Yang, C., Delcher, C., Shenkman, E., &amp; Ranka, S. (2018, November 20). Machine learning approaches for predicting high cost high need patient expenditures in health care - biomedical engineering online. BioMed Central. Retrieved April 22, 2022, from https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0568-3

[4] Rakshit, P., Zaballa, O., Pérez, A., Gómez-Inhiesto, E., Acaiturri-Ayesta, M. T., &amp; Lozano, J. A. (2021, June 14). A machine learning approach to predict healthcare cost of breast cancer patients. Nature News. Retrieved April 22, 2022, from https://www.nature.com/articles/s41598-021-91580-x

Dataset took from Kaggle:
https://www.kaggle.com/datasets/tejashvi14/medical-insurance-premium-prediction