Title
Analysis and Modeling for Ganoderma Data

Author
Wu, Zheqi

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Analysis and Modeling for Ganoderma Data

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Zheqi Wu

2019

ABSTRACT OF THE THESIS

Analysis and Modeling
for Ganoderma Data

by

Zheqi Wu
Master of Science in Statistics
University of California, Los Angeles, 2019
Professor Hongquan Xu, Chair

It is fundamentally challenging to learn from small data sets. In this paper, we analyze ganoderma data, also called Lingzhi, which has a tiny dataset with quite a lot of chemical substances. It is quite a challenge to not only build suitable models fitting the small data but also do the feature extraction to identify the critical subgroup of the chemical substances that are effective to the cancer treatment. This paper does data preprocessing first to adjust the response variable, eliminate outliers and deal with multicollinearity problem. Secondly, we use four datasets with both linear and non-linear models to experiment. It shows that XGboost model has the best fitness of dataset. Also, Principal Component Analysis and Partial Least Square transformation techniques are suitable for our feature dimension reduction purpose that it can reduce the features from 24 dimensions to 5. In the discussion part, we analyze the feature importance between the model with the best performance and the original features.

The thesis of Zheqi Wu is approved.

Frederic R Paik Schoenberg

Ying Nian Wu

Hongquan Xu, Committee Chair

University of California, Los Angeles

2019

This is for my grandma, Liping Xie, who is always happy and eternally young.

Stay passionate, keep curious, and I love you three thousand.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Prof. Hongquan Xu. The door to Prof. Xu office was always open whenever I ran into a trouble spot or had a question about my research or writing. He steered me in the right direction whenever he thought I needed it.

I would also like to thank Prof. Frederic Paik Schoenberg and Prof. Ying Nian Wu for serving as my thesis committee members and providing feedbacks.

And finally, my family, who always have my back.

CHAPTER 1

Introduction

1.1 Motivation

The International Agency for Research on Cancer (IARC) released the latest estimates on the global burden of cancer, which have risen to 18.1 million new cases and 9.6 million deaths in 2018 [JBL18]. In a statistics perspective, men have 20 per cent chance and women have 16.7 per cent chance that develops cancer during their lifetime, and 10 per cent of people die of the disease. It is not hard to hear from our daily life and news about the disease, and the morbidity of cancer and death rate remain high. This phenomenon is due to several factors, including population growth and ageing as well as the changing prevalence of certain causes of cancer linked to social and economic development, and may develop over a long period, even during treatment [BFS18]. That also makes the battle between human and cancer more cost-effective and harder.

In the meanwhile, traditional herbal medicines become a research hotspot in global health debates. In China, traditional herbal medicine has evolved for ages and played a prominent role to address health problems, and it has been fast growth in cancer chemoprevention and therapy area recently. Research regarding the antiproliferative and cytotoxic effects of traditional Chinese medicine (TCM) are being pursued to develop evidence-based complementary and alternative medicine or drug discovery [BFS18], which make it promising that TCM could be a potential approach for cancer treatment. In this paper, we would analyze one particular herb data, called ganoderma, for cancer treatment.

However, medical data sets are usually small and have very high dimensionality. Too many attributes will not necessarily increase accuracy and will make the model too complex to overfit. In the other hand, too few data will make the model less robust. Also, different from western medicine, batch-to-batch variances of herb data are large, which makes the analysis of herb data more challenging.

1.2 Data Description

In this paper, we analyze ganoderma data, also called Lingzhi, provided by a Chinese research institute for medical study. There are real tumor cell experiment records from different ganoderma batches. Since these long term biological experiments are expensive and hard to get, we only have a very small dataset with quite a lot of chemical substances. The main goal of the analysis was to identify a group of substances to predict whether a ganoderma is effective for inhibiting tumor cells. It is becoming more of a challenge to not only build state-of-the-art predictive models with these small data points [JZ97], but also gain an understanding of which substances are really matters in cancer treatments.

In other words, we do not really pursue a super accurate predictive model in this paper, but to identify the critical subgroup of the chemical substances that work to cancer. A sample of data is displayed in Figure 1.1 and Table 1.1. The response variable **Inhibition_Rate** is a continuous variable indicating the treatment power to constrain the growth of tumor cells. The higher value of inhibition rate, the more effective to the cancer cells. If it is negative, it means that this batch of ganoderma does not inhibit the cancer cell growth, but on the other hand, promote cell growth.

We extract a total of 24 features for each batch of Ganoderma, named as **X1**, **X2**, etc., to predict the inhibition rate of tumor cells. The features are the content of different chemical substances in Ganoderma. Individually, each peak in the chromatogram with absolute value less than 0.8 can be considered as the same signal, representing one certain chemical substances, and then we convey the signal

	X1	X2	X3	X4	...	X23	X24	Inhibition_Rate
1	0.000389726	0.001389061	0.001131496	0.00523774	...	0.007242374	0.02135177	94.207299
2	0.000361589	0.001592511	0.000591636	0.005990382	...	0.005943114	0.0160348	-78.071933
3	0.000309676	0.001048389	0.000313446	0.003947481	...	0.006828488	0.02231169	86.313680
4	0.001151718	0.003541977	0.003910359	0.004782069	...	0.004730542	0.006129344	89.665432
5	0.00034104	0.000975084	8.09E-05	0.004596176	...	0.005432924	0.009830344	91.219868

Table 1.1: A Sample of Data

into continuous numbers as 24 columns in the dataset.

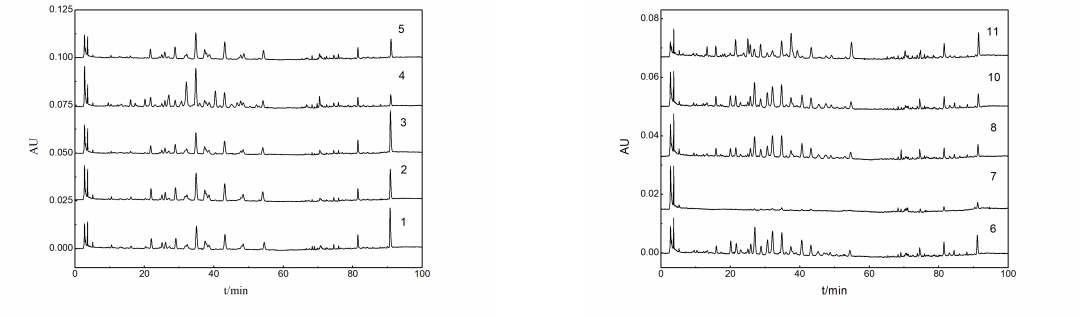


Figure 1.1: Chromatogram of Original Data

1.3 Data Preprocessing

For a given project, there are many factors that would affect the success of data analysis. The representation and quality of the instance data are first and foremost. If there is much irrelevant and redundant information present or noisy and unreliable data, the knowledge discovery during the training phase is difficult. Data preprocessing includes data cleaning, normalization, transformation, feature extraction, and selection, etc [KKP06]. Consider we only have 252 observations in the dataset, we need to be more cautious in the preprocessing step. Better understanding our dataset first is critical for choosing suitable models and improving the model performance in the following steps.

1.3.1 Response Variable

Figure 1.2 shows the distribution of our response variable.

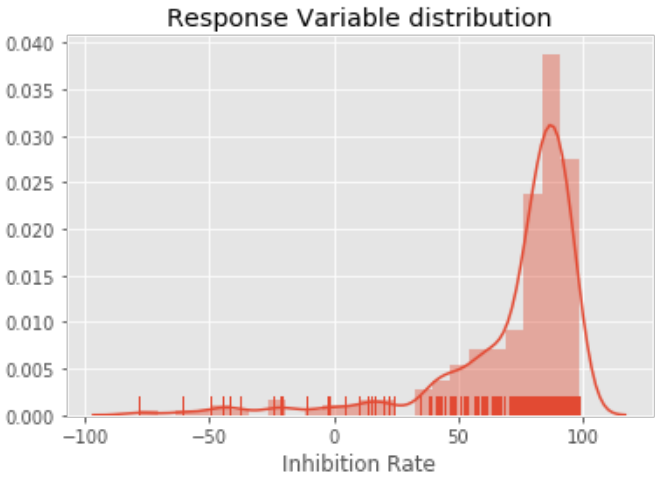


Figure 1.2: Inhibition Rate Distribution

It is highly skewed. The log transformation can be used here to make the distribution less skewed. Consider there are negative values in the response variable, we do the transformation as:

$$\mathrm{Y} = \log(100 - \mathrm{Inhibition\_Rate})$$

The distribution of the response variable after transformation displays in Figure 1.3. It is nearly normally distributed compared to the previous one. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics.

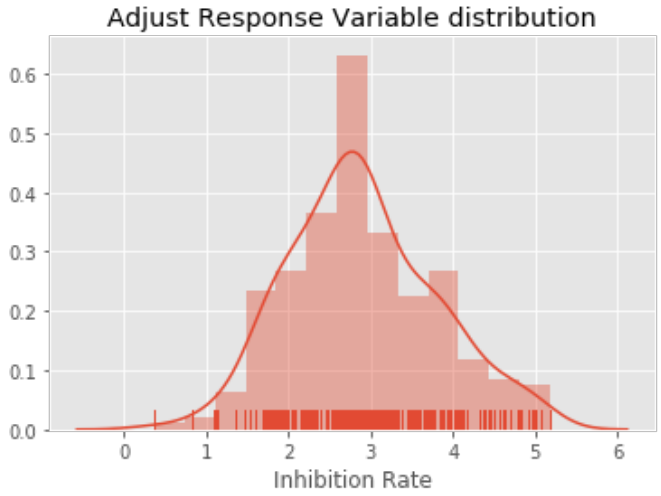


Figure 1.3: Adjust Response Variable Distribution

1.3.2 Outliers

Since these data were from biological experiments, we may have some outlier/extreme values from the experiments that would have a huge potential impact on our model fitting and parameter tuning [CC10]. Treating or altering the outlier in genuine observations is not a standard operating procedure. However, it is essential to understand their impact on our predictive models.

Our 252 observations of ganoderma data were collected from two experiments, that is to say, we actually only have 126 different types of ganoderma, but the experimental researchers repeated the experiments by two different technicians. So we decided to use them both. Furthermore, we can compare these two responses using scatter plot to decide whether there are outliers or not. Look at the red points No.2 and No.10 in Figure 1.4, they are far away from most of the data and should be excluded from such model fitting. We decide to use the median value to replace these values of outliers in the response variable.

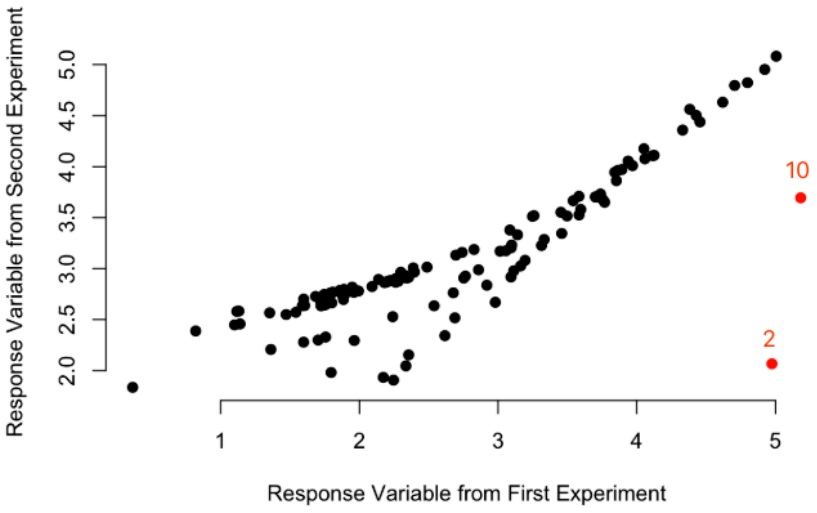


Figure 1.4: Scatter Plot

1.3.3 Multicollinearity

Next, we would like to see the correlation matrix between each column in the dataset.

Two variables are collinear if there is a linear relationship between them. Multicollinearity involves more than two variables. In Figure 1.5, white color squared boxes means strong multicollinearity here in the plot. For example, the correlation between X4 and X12 is 0.93, which is a nearly perfect match. If existing multicollinearity problem, regression estimates are unstable and have high standard errors. It does not mean that we cannot use the linear regression models in the prediction; however, we are not sure that which feature contributes more to the result in the feature selection part.

In Section 2.1, we will use some data transformation techniques to deal with the multicollinearity problem for our dataset, such as principal component analysis (PCA) and partial least squares (PLS).

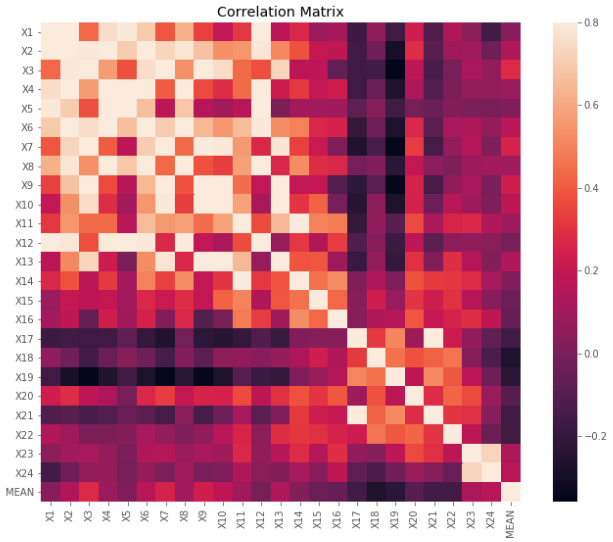


Figure 1.5: Correlation Matrix

CHAPTER 2

Methodology

2.1 Data Transformation

Our dataset is relatively small with 252 observations and 24 features, some of which have multicollinearity problems. Too many data features always affect the efficiency of the analysis, and sometimes affect modeling accuracy. Therefore, how to use the feature extraction method to select an effective subset of attributes, or features, is an important question. The commonly used methods include principal component analysis (PCA), independent component analysis (ICA), partial least squares (PLS) , canonical correlation analysis (CCA) , and genetic programming (GP) [LLH11]. Both principal components analysis and partial least squares regression produce factor scores as linear combinations of the original predictor variables so that there is no correlation between the factor score variables used in the predictive regression model so that they can deal with the multicollinearity of the features. We will try both dimension reduction methods for our dataset in the following chapter.

2.1.1 Principal Component Analysis

Principal component analysis (PCA) is a widely used technique in dimensionality reduction area [WEG87]. It looks for a few orthogonal linear combinations of the variables such that the maximum variance is extracted from the variables. It can be used to summarize the data without losing too much information in the process [Shl14]. Mathematically, the principle components (PCs) are obtained by eigen decomposition of the covariance or correlation matrix of the dataset. The first PC,

s_1, is the linear combination with the largest variance. We can form it as:

$$s_1 = x^T w_1,$$

where $x = (x_1, x_2, \ldots, x_p)$ is our dataset and $w_1 = (w_{11}, w_{12}, \ldots, w_{1p})^T$ is the p-dimensional coefficient vector solves

$$w_1 = \text{argmax}_{||w||=1} \text{Var}\{x^T w\}.$$

The second PC is the linear combination with the second largest variance that is orthogonal to the first PC, and so on. There are as many PCs as the number of original variables, which is 24 for our dataset.

Look at Figure 2.1. In our case, we already have 79% information/explained variance when we have the first five components. And up to 10 PCs, we will have 93.6% information compared to the original dataset with 24 features.

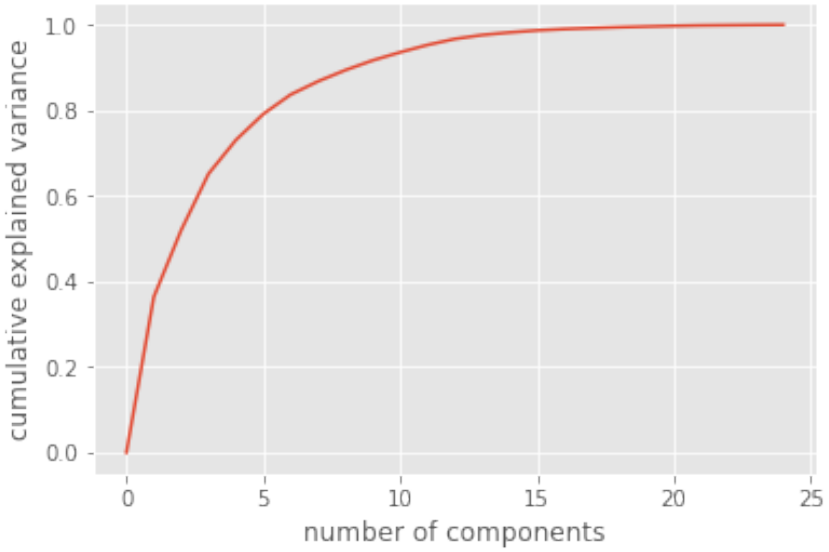
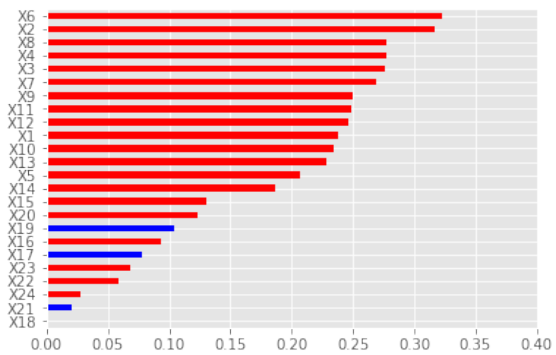


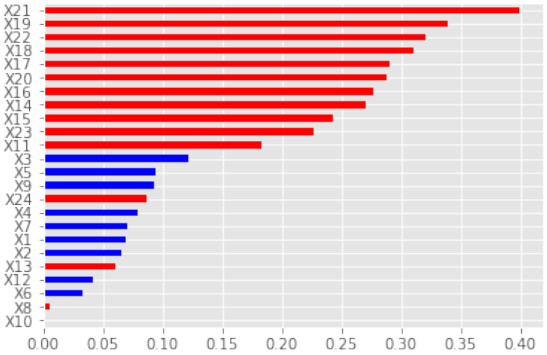
Figure 2.1: PCA Cumulative Explained Variance Ratio

The full data are too numerous to quote here, nor are they sufficiently interesting to cite in full. The first five principal eigenvectors, are listed in Figure 2.2. Also, the sign of PCA eigen-vector can be positive or negative. We assume that there is a positive direction, as shown in red in Figure 2.2 and blue bar means it towards

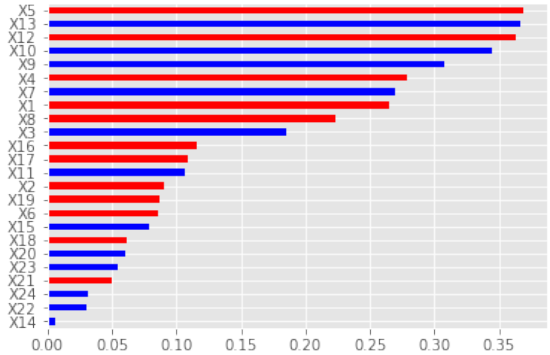
the negative direction. Unfortunately, we cannot eliminate some features by the construction of the first five PCA components. Ideally, if the PCA components are mostly constructed by some specific features and we can assume others are redundant and then drop them. However, in our case, the first PC is consist of features X1 to X13 and the rest composed of the second PC except feature X24, if we set the threshold to be 0.2, and the values are all positive. For third to fifth PCs, they are less important but also very interesting. It shows that the features consisting of the third PC are similar to the first PC, but there are some negative values in it, such as X13, X10, X9, and X7. The fourth and fifth PCs are more like the second PC, also with negative values.



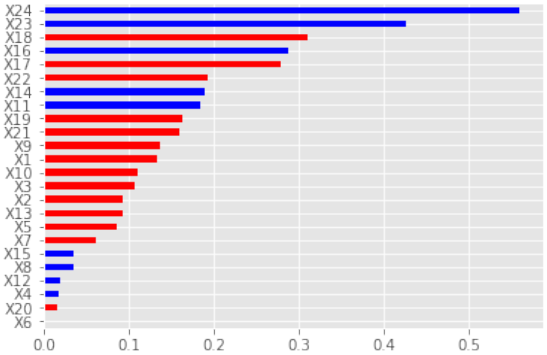
(a) PCA1



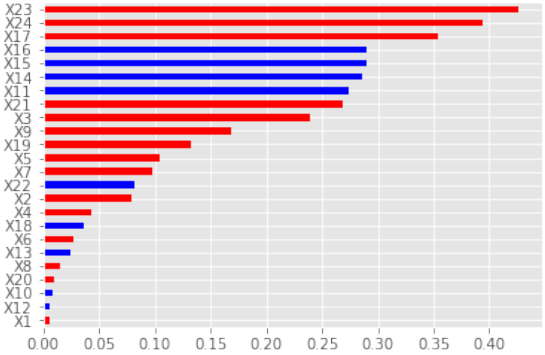
(b) PCA2



(c) PCA3



(d) PCA4



(e) PCA5

Figure 2.2: First Five Principle Eigenvectors

2.1.2 Partial Least Square

One drawback of the PCA technique is that it captures only the characteristics of the X-vector or predictive variables [BS06]. It is impossible to estimate how each predictive variable may be related to the dependent or the target variable. In a way, it is an unsupervised dimension reduction technique [MY08]. There may be a considerable improvement if we use a technique not only to capture as much information in the raw predictive variables but also in the relation between the predictive and target variables. Partial least square (PLS) allows us to achieve this balance [RK05].

Assume X is an $n \times p$ matrix and Y is an $n \times 1$ matrix. PLS technique tries to find a linear decomposition of X and Y such that $X = TP' + E$ and $Y = UQ' + F$, where

$$T_{n\times r} = X\_\textbf{scores} \quad U_{n\times r} = Y\_\textbf{scores}$$

$$P_{p\times r} = X\_\textbf{loadings} \quad Q_{1\times r} = Y\_\textbf{loadings}$$

$$E_{n\times p} = X\_\textbf{residuals} \quad F_{n\times 1} = Y\_\textbf{residuals}$$

Decomposition [Kit96] is finalized so as to maximize covariance between T and U. There are multiple algorithms available to solve the PLS problem such as non-linear iterative partial least squares (NIPALS) and SIMPLS algorithm. However, all algorithms follow an iterative process to extract the X**_scores** and Y**_scores**.

Note that the PLS algorithm automatically predicts Y using the extracted Y**_scores** (U). However, we do not want the transformed Y and the goal is to obtain the X**_scores** (T) from the PLS decomposition and use them separately for a regression to predict Y. This provides us the flexibility to use PLS to extract orthogonal factors from X while not restricting ourselves to the original model of PLS.

Similar to PCA, we list the loadings here for the first five PLS transformation in Figure 2.3. In the feature selection perspective, the results are better than PCA. If we set the threshold equal to 0.2, some features have never been selected for the

first five PCs, such as **X3, X10, X12, X14, X15**, so that we can treat them as redundant features.

2.2 Stepwise Regression Model

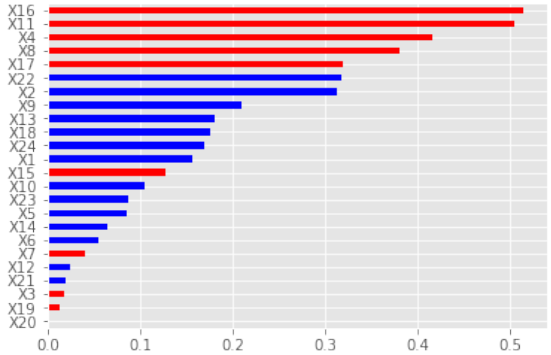
Stepwise regression is the only regression technique that would be introduced in this paper. It builds a model by adding or removing the predictor variables, generally via a series of T-tests or F-tests. The variables, which need to be added or removed, are chosen based on the test statistics of the coefficients estimated.

Stepwise regression requires four assumptions that are the same as other multiple regression models. First, it must be a linear relationship between predictive variables and the response variable. Second, the residuals are normally distributed. Third, it assumes that there is no multicollinearity in the data. The last one is the homoscedasticity.

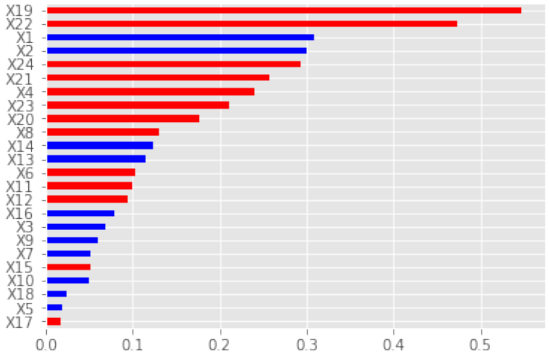
Since the complexity of our medical dataset, we doubt that there is a linear relationship between X and Y to satisfy the first assumption. Therefore, we set the stepwise regression as our base model and we will try random forest and XGboost later to exam the non-linear relationship of our dataset. However, after data preprocessing steps, our dataset does meet the other assumptions for the model and we can expect to have a not so bad result. See plots below.

Specifically, we use both backward elimination and forward selection for the model, and we set the degree equals to 2 which means the model would automatically adding quadratic forms and interactions between two features. The fitted model is too long to be shown here. So we plot the residual diagnostic below.

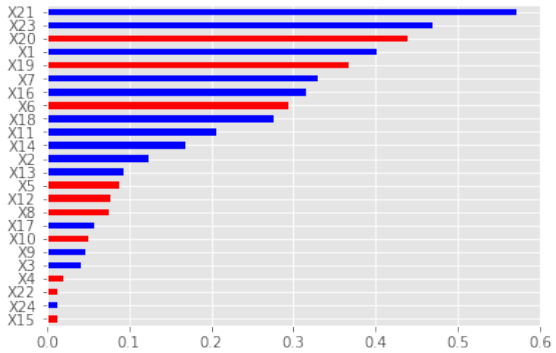
Residuals vs fitted plot can check the last assumption. Figure 2.4 (top) shows that the variance of residuals decreases as the fitted values increase, suggesting somehow heteroscedastic. Q-Q plot in Figure 2.4 (bottom) can check the second assumption. We see that most of the points falling along a straight line in the Q-Q plot, which provides strong evidence that the residuals of the regression are



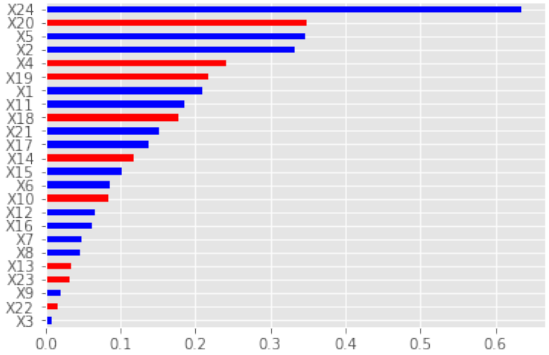
(a) PLS1



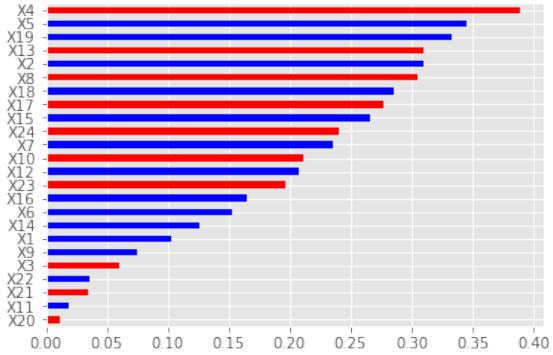
(b) PLS2



(c) PLS3



(d) PLS4



(e) PLS5

Figure 2.3: First Five PLS loadings

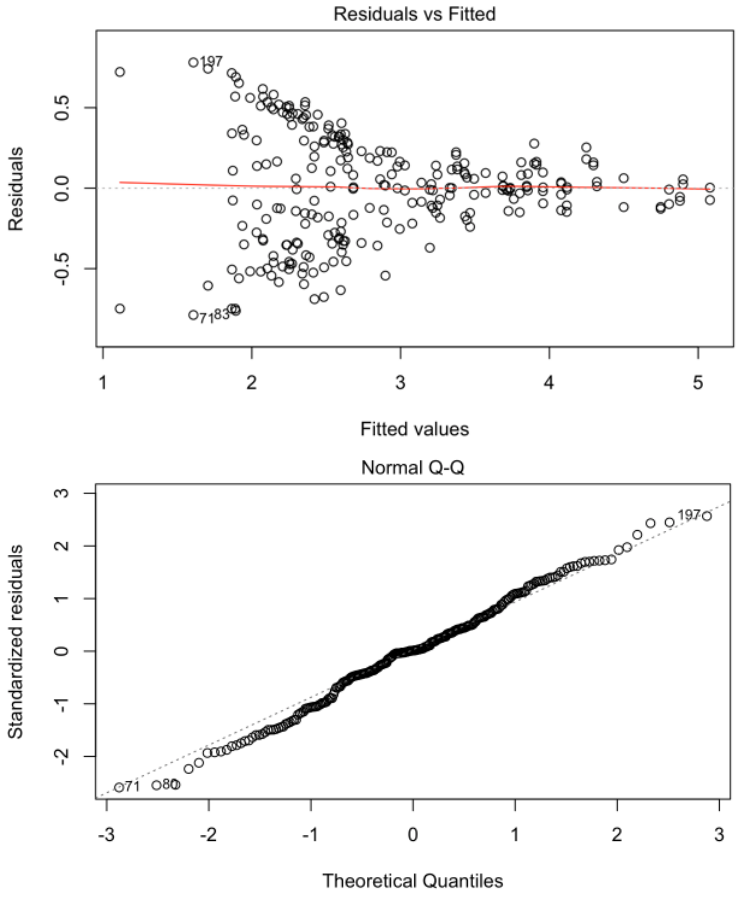


Figure 2.4: Stepwise Regression Residual Diagnostic

normally distributed.

Also, we would like to exam the outliers for this particular linear model. We first use Cook's distance to decide if the data point is an outlier. The Cook's distance for each observation i measures the change in $\hat{\mathbf{Y}}$ (fitted Y) for all observations with and without the presence of observation i, so we know how much the observation i impacted the fitted values. Mathematically, Cook's distance D_i of observation i (for $i = 1, ..., n$) is computed as:

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{y}_j - \hat{y}_{j(i)}\right)^2}{p \times s^2}$$

where $\hat{y}_j$ is the fitted value of observation j using all the observations, $\hat{y}_{j(i)}$ is the fitted response value obtained when excluding i in training process, n is the number of observations, p is the number of features for each observation and s^2 is the mean

squared error of the regression model.

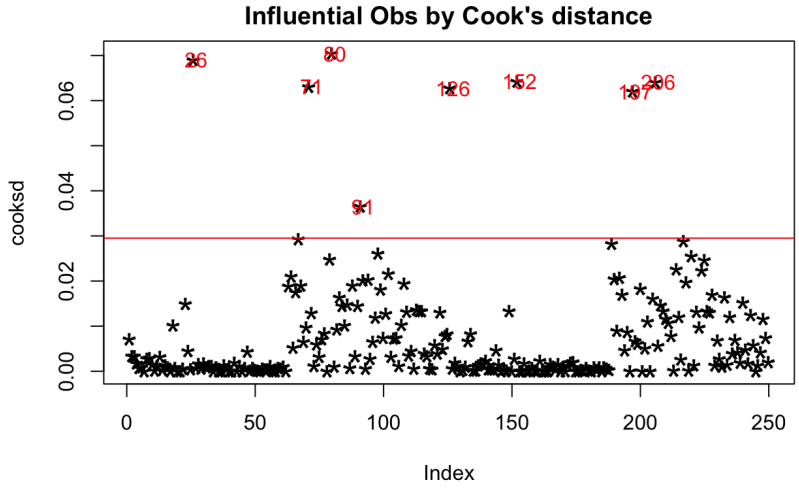


Figure 2.5: Influential Observations by Cook's distance

In general use, those observations that have a Cook's distance greater than four times the mean may be classified as influential, that is 0.02951547 in our case, plotted as a red line in Figure 2.5. In particular, there are several Cook's distance values (No.26 and No.80) that are relatively higher than the others, which exceed the threshold value. We want to find and omit these from our data and rebuild models.

Same as Section 1.3, we use the median value to replace these value of outliers in the response variable. We will fit those two datasets with/without outliers into models in the following part.

2.3 Random Forest Model

Random forest is an ensemble method for regression or classification. Given a training dataset, it constructs multiple decision trees based on bootstrapping, and each decision tree is trained using a random subset of all the features. Random forest has been shown to prevent overfitting and reduce the variance.

During the generation of bootstrapped samples (i.e., tree bagging), given a train-

ing feature matrix $X = [x_1, x_2, \cdots, x_n]^T$, where n is the number of observations, and each x_i is p-dimensional feature vector, and a response vector $Y = [y_1, y_2, \cdots, y_n]^T$, bagging will repeat B times to sample with replacement of the n observations and subsamples without replacement of the p features and construct a decision tree based on the bootstrapped training data [LW02]. The detailed algorithm is shown below:

(1)Draw B bootstrap samples from the original data.

(2)For each bootstrap sample, grow an unpruned regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample p_0 $(0 < p_0 < p)$ of the p predictors and choose the best split from among those variables.

(3)Predict new data by averaging the predictions of the B trees. Remark that the algorithm has the particularity of exploiting two layers of randomness, random inputs (bootstrap) and random features (random selection of a subset of predictors), which significantly enhances its speed and accuracy.

As we all know, the key to improving the performance of those machine learning models is hyper-parameter tuning. In our random forest model, we used GridSearch package using three folds cross-validation and tuned three hyper-parameters listed below:

max_depth: the maximum depth of the individual tree, reduction of the maximum depth helps fighting with overfitting.

max_features: the maximum number of features random forest is allowed to try in an individual tree.

n_estimators: the number of trees we want to build before averaging the predictions. Higher number of trees gives better performance and makes the predictions stronger and more stable.

For example, for our dataset eliminating the outliers, the best tuning hyper-parameters are {'**max_depth**': 14, '**max_features**':8, '**n_estimators**': 100}.

2.4 XGboost Model

eXtreme Gradient Boosting (XGBoost) is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the gradient boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way [CG16]. It runs fast and can solve problems beyond billions of examples. It is well known as a highly flexible and versatile tool that can work through most regression, classification and ranking problems as well as user-built objective functions.

It is also a tree-based model using gradient boosting machine. The detailed algorithm is shown below:

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x)) = (y - F(x))^2$ for regression, and number of iterations M.

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

2. For $m = 1$ to M, let $F_{m-1}(X)$ be the prediction at $(m-1)$th step.

2.1 Compute so-called pseudo-residuals r_{im} for i observations in m-th step:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = 2(y_i - F_{m-1}(x_i)).$$

2.2 Fit a base learner (tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $(x_i, r_{im})_{i=1}^n$.

2.3 Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

where γ is a lagrangian multiplier to control the model complexity, also known as regularization term in the loss function.

2.4 Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Same as random forest, we should tune the hyper-parameters for our model to improve the power of the XGboost getting better prediction. Here we used three folds cross-validation by GridSearch, and chose our hyper-parameters listed below:

max_depth: Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.

gamma: Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be.

min_child_weight: Minimum sum of instance weight (hessian) needed in a child. The larger **min_child_weight** is, the more conservative the algorithm will be.

subsample: Subsample ratio of the training instances. This would prevent overfitting.

n_estimators: the number of trees we want to build before averaging the predictions. Higher number of trees gives better performance and makes the predictions stronger and more stable.

For example, for our dataset eliminating the outliers, the best tuning hyper-parameters are {'**max_depth**': 5, '**gamma**':0.1, **min_child_weight**:10, **subsample**: 0.6, '**n_estimators**': 100}.

CHAPTER 3

Results

3.1 Coefficient of Determination

Coefficient of Determination ($\mathbf{R^2}$) is a measure used to check a model's goodness of fit. R-squared has the useful property that its scale is intuitive: it ranges from zero to one, with zero indicating that the proposed model does not improve prediction over the mean model, and one indicating perfect prediction. Improvement in the model results in proportional increases in R-squared. Mathematically, it is defined as:

$$\mathbf{R^2} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}$$

where $\hat{y}_i$ is our individual predicted value, y_i is observed response value and $\bar{y}$ is the mean of y_i.

3.2 RMSE

The Root Mean Square Error (RMSE) is the standard deviation of the residuals. It indicates how spread out these residuals are – how close the observed data points are to the model's predicted values. Whereas R-squared scaled between 0 and 1, which is a relative measure, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. It is defined as:

$$\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

where $\hat{y}_i$ is our individual predicted value, y_i is observed response value and n is the number of observations.

3.3 Comparison

In this paper, we compared four different datasets with stepwise regression, random forest and Xgboost methodology, and we record the R-squared results in Table 3.1 and RMSE results in Table 3.2. Here is the explanation for the four datasets:

a. **Original dataset**: the original dataset with $\mathbf{log}(\mathbf{Y})$ transformation.

b. **Outlier dataset**: the dataset replacing outliers from **original dataset**.

c. **PCA dataset**: the dataset using PCA transformation from **outlier dataset**.

d. **PLS dataset**: the dataset using PLS transformation from **outlier dataset**.

Also, we only write down the highest value of R-squared or lowest RMSE with specific number of components for each model in Table 3.1 or Table 3.2. The p in tables stand for the number of components achieving the best value.

	original	outlier	PCA	PLS
Stepwise Regression	0.8201	0.8269	0.8328 (p=24)	0.8331 (p=24)
Random Forest	0.8106	0.8158	0.8232 (p=23)	0.8143 (p=21)
XGBoost	0.8417	0.8505	0.8435 (p=14)	0.8432 (p=16)

Table 3.1: R-squared Table

	original	outlier	PCA	PLS
Stepwise Regression	0.3781	0.4922	0.3535 (p=24)	0.3530 (p=24)
Random Forest	0.3776	0.3711	0.3636 (p=23)	0.4301 (p=21)
XGBoost	0.3584	0.3703	0.3452 (p=14)	0.4000 (p=16)

Table 3.2: RMSE Table

Look at Table 3.1 and Table 3.2. First, we can do the horizontal comparison. The **outlier dataset**, **PCA dataset** and **PLS dataset** all perform slightly better than the **original dataset**. It makes sense that for a small dataset, the outlier/extreme value would have a huge impact on the model fitting, and potential effect of hyper-parameter tuning. It is necessary to spend time understanding our dataset and dealing with the outlier problem.

Furthermore, in the stepwise regression model, the datasets after PCA or PLS transformation have a larger R-squared value than original dataset from 0.82 to 0.83. As we discussed above, the stepwise regression assumes there is no multicollinearity in the data. So after doing the orthogonal transformation of the dataset, they performed better than before.

Secondly, let us do the vertical comparison. The Xgboost model performs best with maximum of R squared value of 0.8505 and minimum of RMSE value of 0.3452. It is surprising that the random forest model can not beat our base stepwise regression. It may due to smallness of the dataset or unstable of random forest.

Thirdly, if we compare the results from PCA and PLS datasets, PLS datasets

have higher value of R squared (0.8328 to 0.8331)and smaller value of RMSE (0.3535 to 0.3530) for stepwise regression model. However, it do not work for non-linear models like random forest and Xgboost.

Besides, Figures 3.1 and 3.2 record R-squared value or RMSE value with PCA components from 1 to 24. Figures 3.3 and 3.4 record those values with PLS components from 1 to 24.

Look at Figure 3.1 to 3.4 we can conclude that PCA and PLS transformation is suitable for our feature dimension reduction purpose. For example, the R-squared line for XGboost model with PCA data nearly flatten at five or more components. It eliminates the redundant features from 24 to 5 but still maintains a great value of R-squared and RMSE.

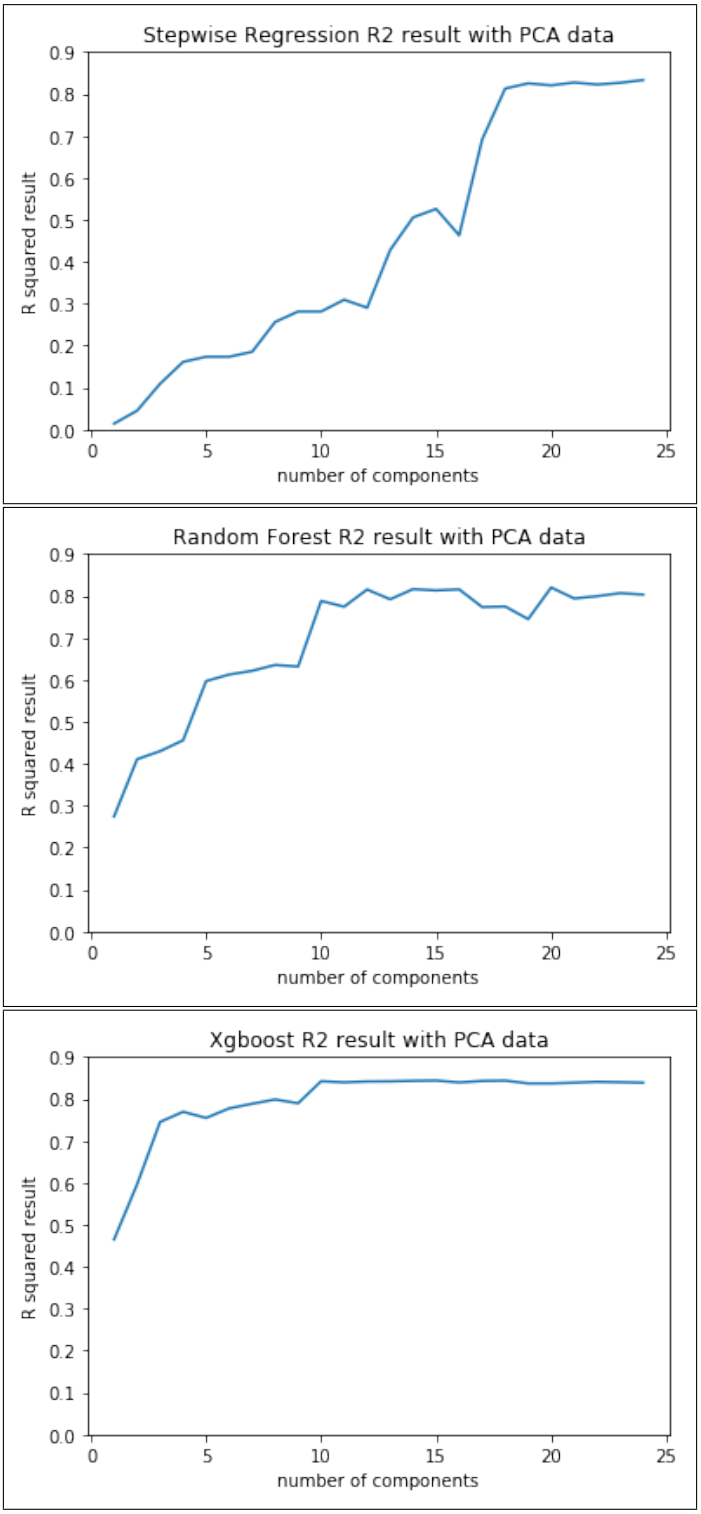


Figure 3.1: R-squared with PCA data from different models

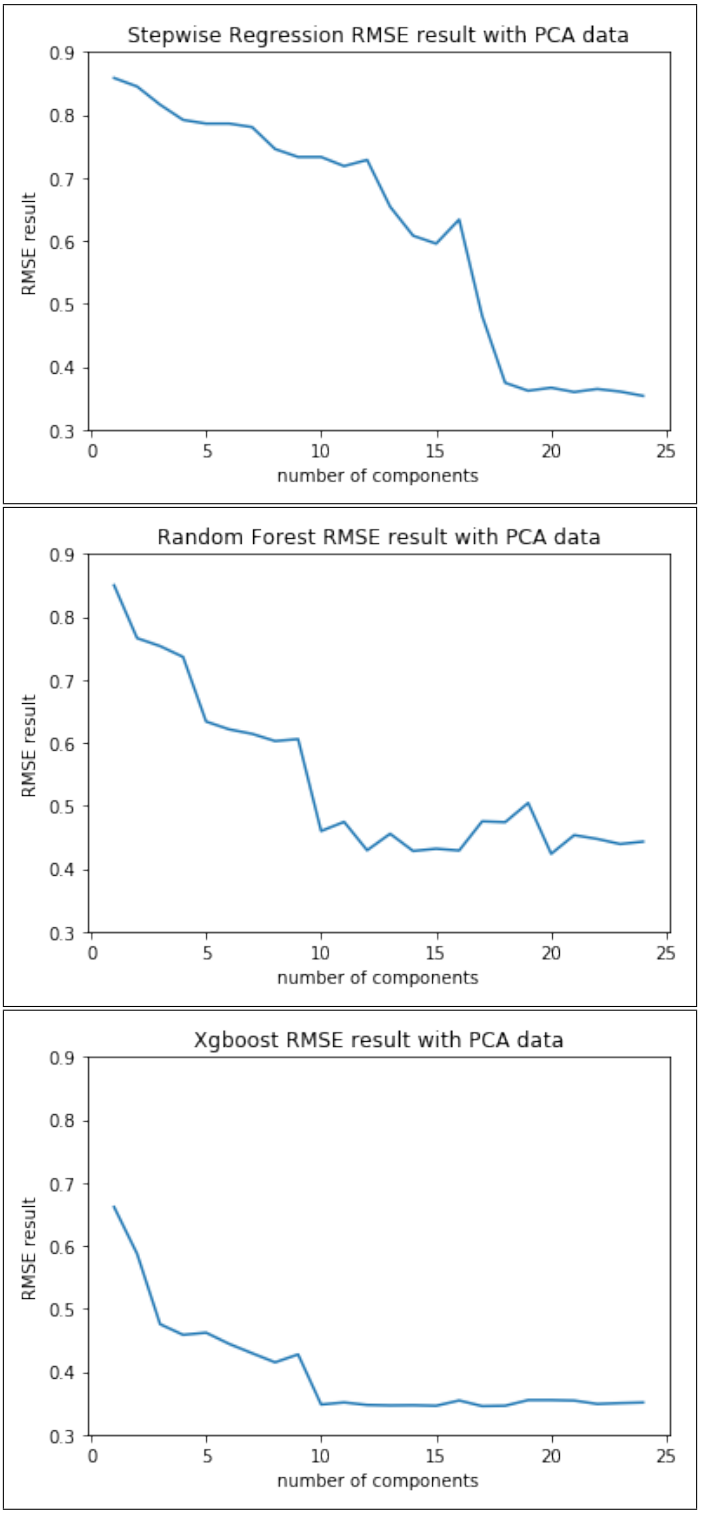


Figure 3.2: RMSE with PCA data from different models

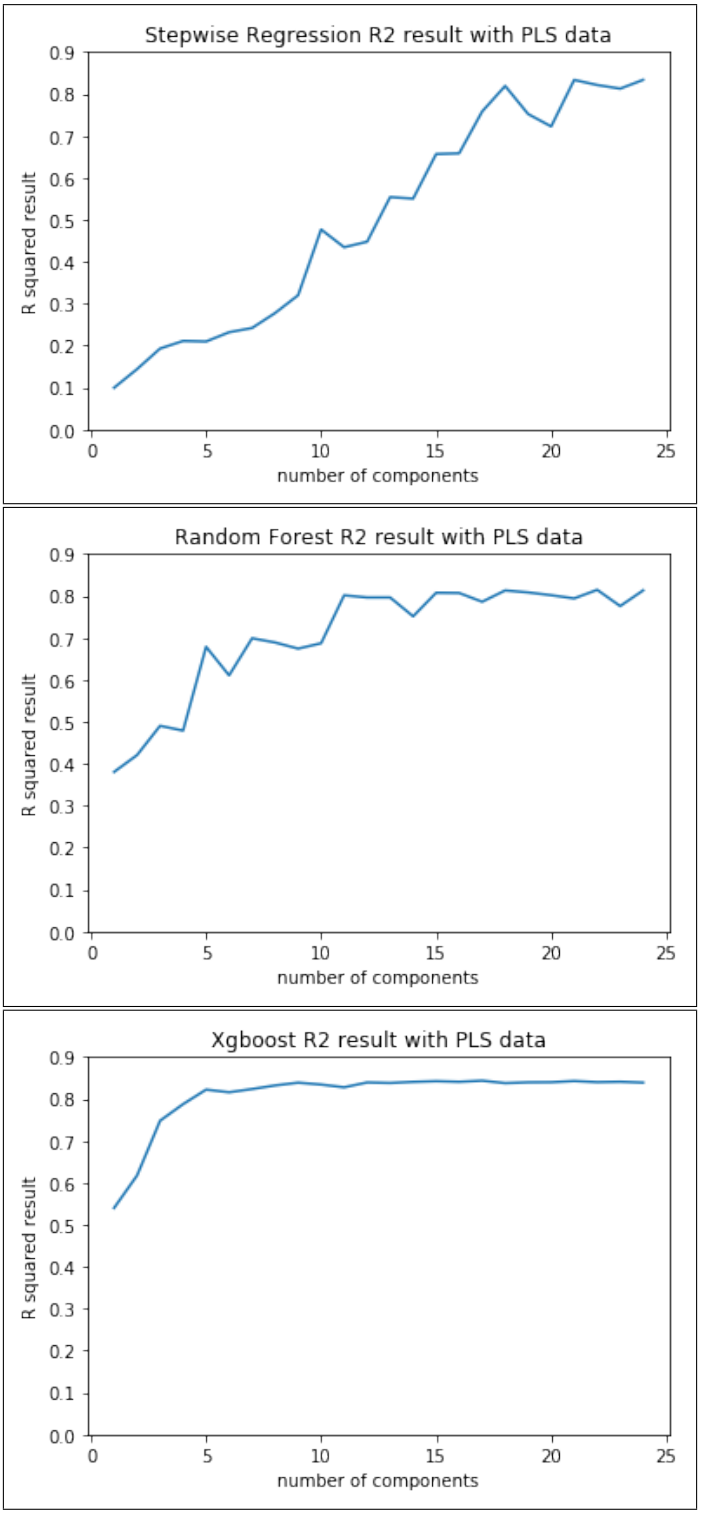


Figure 3.3: R-squared with PLS data from different models

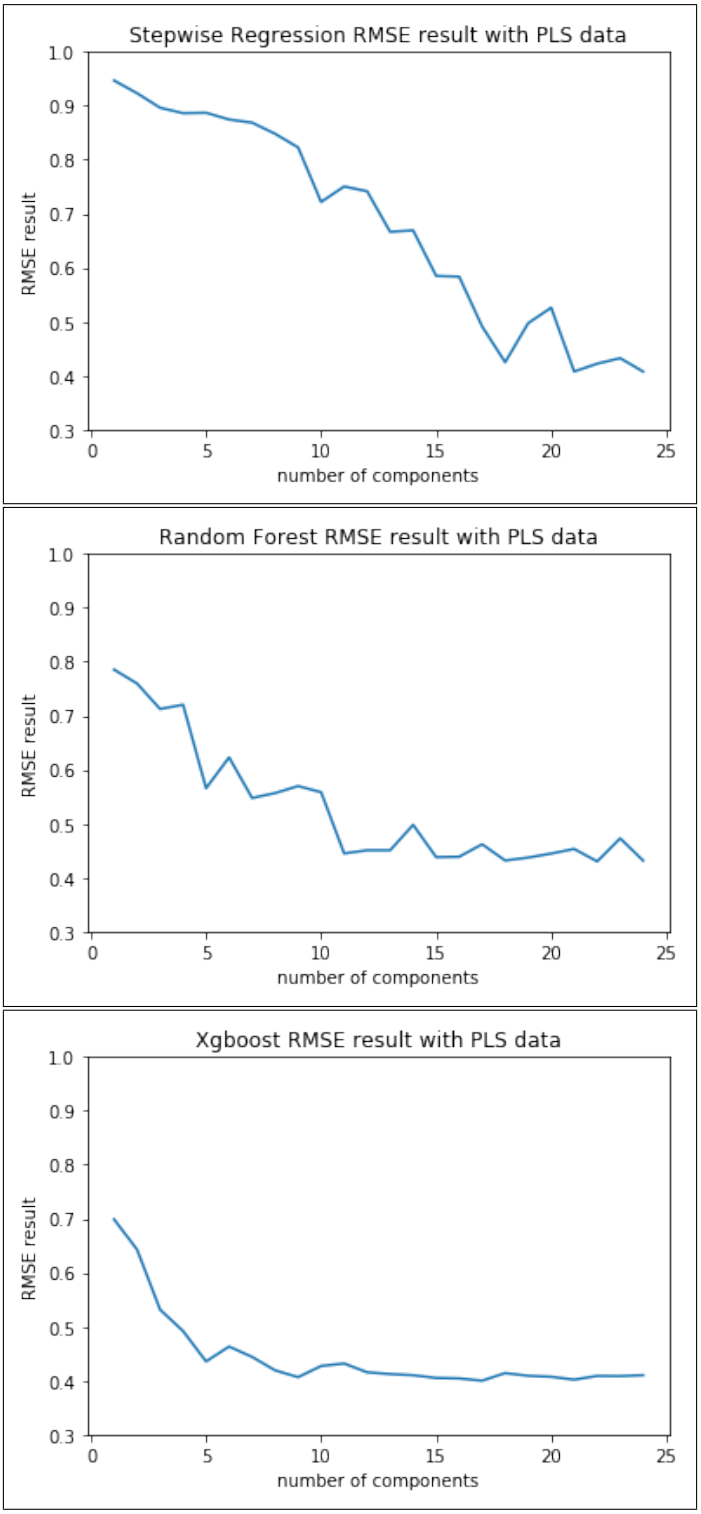


Figure 3.4: RMSE with PLS data from different models

CHAPTER 4

Discussion and Conclusion

We have two goals for analyzing our ganoderma data, not only to get a great prediction which means a good R squared or RMSE value but also need to choose the features that matter. It is hard to tell because their relationship is quite complex and hard to separate the effect even using PCA or PLS transformation. As we mentioned above, we can fit the random forest and XGboost model using the first five to ten PCs or PLSs. But we would like to discuss the feature importance between the model and the original features here.

Partial dependence plot (PDP) shows the marginal effect features have on the predicted outcome of a machine learning model [Fri01]. A partial dependence plot can show whether the relationship between the target and a feature is linear, monotonous or more complex.

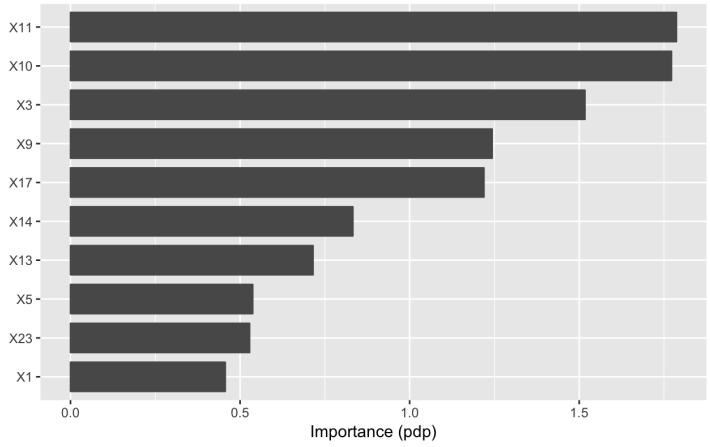


Figure 4.1: Feature Importance

Figure 4.1 lists the first ten important original features in the XGboost model

with best tuning parameters. All of them are part of construction in the first two PCs or PLSs.

Medical data sets are usually small and include a large number of attributes. When the data set is small, there is a high level of uncertainty. The insufficient and incomplete information in small data sets causes incorrect analysis. This research analyzes the ganoderma data, compares four datasets to figure out the impact of outliers and data transformation methods on linear and non-linear models. It shows that Xgboost has highest R squared value and minimum RMSE which has the best fitness of the data. Also, PCA and PLS transformation are suitable for our feature dimension reduction purpose that it can reduce the features from 24 dimensions to 5.

REFERENCES

[BFS18] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L. Siegel, Lindsey A. Torre, and Ahmedin Jemal. "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries." *CA: A Cancer Journal for Clinicians*, **68**(6):394–424, 2018.

[BS06] Anne-Laure Boulesteix and Korbinian Strimmer. "Partial least squares: a versatile tool for the analysis of high-dimensional genomic data." *Briefings in Bioinformatics*, **8**(1):32–44, 05 2006.

[CC10] Denis Cousineau and Sylvain Chartier. "Outliers detection and treatment: a review." *International Journal of Psychological Research*, **3**(1):58–67, 2010.

[CG16] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, 2016.

[Fri01] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine." *Annals of statistics*, pp. 1189–1232, 2001.

[JBL18] Lijing Jiao, Ling Bi, Yan Lu, Qin Wang, Yabin Gong, Jun Shi, and Ling Xu. "Cancer chemoprevention and therapy using chinese herbal medicine." *Biological Procedures Online*, **20**(1):1, Jan 2018.

[JZ97] Anil Jain and Douglas Zongker. "Feature selection: Evaluation, application, and small sample performance." *IEEE transactions on pattern analysis and machine intelligence*, **19**(2):153–158, 1997.

[Kit96] Barbara A Kitchenham. *Software metrics: measurement for software process improvement*. Blackwell Publishers, Inc., 1996.

[KKP06] SB Kotsiantis, Dimitris Kanellopoulos, and PE Pintelas. "Data preprocessing for supervised leaninging." *International Journal of Computer Science*, **1**(2):111–117, 2006.

[LLH11] Der-Chiang Li, Chiao-Wen Liu, and Susan C. Hu. "A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets." *Artificial Intelligence in Medicine*, **52**(1):45 – 52, 2011.

[LW02] Andy Liaw, Matthew Wiener, et al. "Classification and regression by random forest." *R news*, **2**(3):18–22, 2002.

[MY08] Saikat Maitra and Jun Yan. "Principle component analysis and partial least squares: Two dimension reduction techniques for regression." *Applying Multivariate Statistical Models*, **79**:79–90, 2008.

[RK05] Roman Rosipal and Nicole Krämer. "Overview and recent advances in partial least squares." In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*, pp. 34–51. Springer, 2005.

[Shl14] Jonathon Shlens. "A tutorial on principal component analysis." *arXiv preprint arXiv:1404.1100*, 2014.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis." *Chemometrics and intelligent laboratory systems*, **2**(1-3):37–52, 1987.