DATA MINING

Team Name: Easy to Analyze

Niloofar Fadavi, 48073120, CS7331

Sebastian Lu, 48030051, CS7331

Yichien Chou, 48068284, CS7331

Introduction

In order to understand our data, we should look at each variable and try to understand their meaning and relevance to the problem. However, the background of the dataset is not given. In other words, it is hard for us to go deeper without the data and variables descriptions. Though we can't tell the story about the data, our goal is clear in this project: predict RESPONSE using other 37 variables (f1-f37). In this project, we will use different methods (Linear Regression, LASSO, Ridge Regression, Elastic Net, SVM, Random Forest and Neural Network) to train the data and build models for prediction. To pick the best model for prediction, we evaluate models using Mean Squared Error (MSE) and define the best model with the smallest MSE. Then we make predictions using newly given variables on the best model.

First of all, let's learn the dataset we will use in this project:

Training set: 250 observations of 38 variables.

Validation set: 100 observations of 38 variables.

Testing set: 150 observations of 38 variables.

The whole dataset was split into three smaller datasets: training, validation and testing dataset. There are 250 observations in training dataset, 100 observations in validation dataset and 150 observation in testing dataset. In other words, the whole dataset was split at a ratio of 50%/20%/30% refers to training/validation/testing. Overall, there are 500 observations in the whole dataset. There are 38 variables (f1-f37, Response) in training and validation dataset.

Notice that, thought testing set has 38 variables as well, besides f1-f37, there is an ID variable rather than Response variable.

We can't judge whether each variable with correct data type because of lack of data descriptions. All 38 variables (f1-f37, RESPONSE) are numeric variables, and we assume all those are continuous numbers. The ID in testing dataset is not the variable we need to concern; it only represents the index number.

We will use training dataset to build models and validation dataset to test the performance of models using MSE metric and testing dataset to make prediction of RESPONSE.

Data Pre-processing

Before we build models on training set, it is important to clean the dataset.

We checked if there were any missing values in dataset by using sapply() and is.na() function in R. The result is as below:

f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
0	0	0	0	0	0	0	0	0	0
f11	**f12**	**f13**	**f14**	**f15**	**f16**	**f17**	**f18**	**f19**	**f20**
0	0	0	0	0	0	0	0	0	0
f21	**f22**	**f23**	**f24**	**f25**	**f26**	**f27**	**f28**	**f29**	**f30**
0	0	0	0	0	0	0	0	0	0
f31	**f32**	**f33**	**f34**	**f35**	**f36**	**f37**	**response**		
0	0	0	0	0	0	0	0		

There is no missing value in training set.

Then, we checked if there were any duplicated rows or near-zero variance features using duplicated() function and nearZeroVar() function in R.

Missing values, duplicated rows and near-zero variance features are checked. However, none of those values have been detected so we did nothing on training dataset at this step for cleaning purpose.

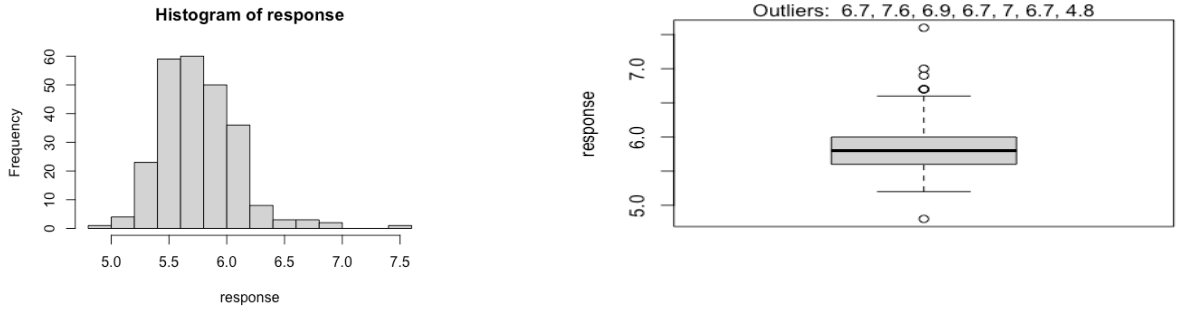
Explore the training dataset and further cleaning

We start exploring the dataset from the 'RESPONSE' variable first which is the variable as the target to predict.

Below is the summary of 'RESPONSE':

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.800	5.600	5.800	5.803	6.000	7.600

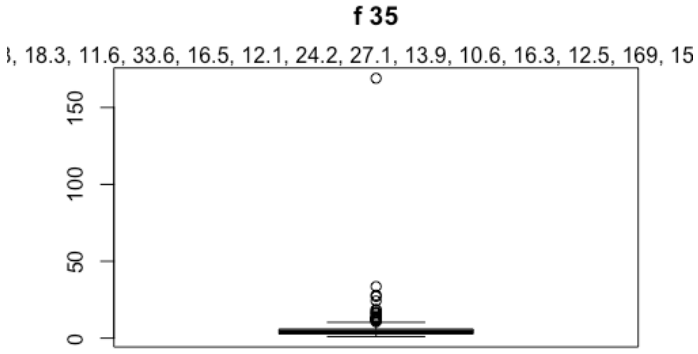
Figure: Histogram and Boxplot of response



The RSEPONSE with minimum value of 4.8 and maximum value of 7.6. The mean and median of RESPONSE are close which 5.803 and 5.800 respectively. We observe the normal distribution

from histogram and some outliers from boxplot. The is no extreme values in RESPONSE variable.

Figure: Boxplot of f35



(All 37 boxplots are in appendix)

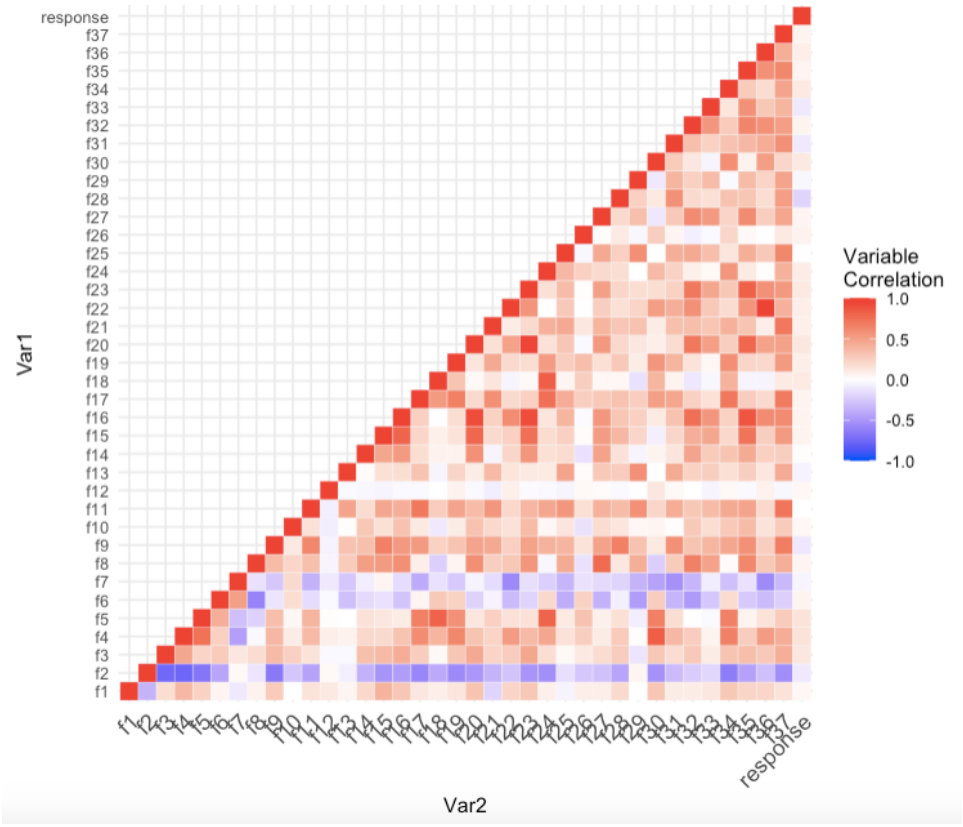
Then we look on all other features (f1-f37). we can look at each variable and try to understand their meaning and relevance to this problem. I know this is time-consuming, but it will give us the flavor of our dataset. Again, due to lack of description about the dataset, we learn features based on distributions and make some reasonable assumptions. We explore all 37 features by exploring corresponding boxplots. We noticed that in some features there are some extreme values in outliers. Although there is no strict or unique rule whether outliers should be removed or not from the dataset before doing statistical analyses, it is quite common to, at least, remove or impute outliers that are due to an experimental or measurement error (like the weight of 1000 kg for a human). Some statistical tests require the absence of outliers in order to draw sound conclusions, but removing outliers is not recommended in all cases and must be done with caution. Overall, we keep outliers in our dataset since we will build different statistical models on the training dataset. However, we remove all those observations looks like experimental or measurement errors which will impact the models we build later. Because we don't have descriptions about data so we cannot set reasonable ranges according to description and we define "errors" by observing the extreme values from boxplot of each feature. For example, from boxplot of f35, there is one observation over 150 and all other observations are smaller than 50, we define this observation (150) as an experimental or measurement error. We manually check all boxplots of features and remove all "errors" as we defined above.

After removing those "errors", 243 observations left in our training dataset which means observations were treated as "errors" and were removed from training dataset. No change in number of variables.

Correlation matrix

Before we build models, we explore the correlations between each two variables. This can give us some ideas about the relationships between variables.

Figure: Correlation plot



According to the correlation heat matrix, most features (f1-f37) show the light color with RESPONSE. This means a weak positive or negative relationship between RESPONSE and another feature. Some even has white color which represents no relationship between two variables.

Normalization

The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. However, normalization is not always needed. First, normalization depends on the algorithm. For some algorithms, normalization has no effect. Generally, algorithms that work with distances tend to work better on normalized data, but this doesn't mean the performance will always be higher after normalization. Secondly, if attributes already have a meaningful and comparable scale then normalization can destroy important information. Take data coming from a physical experiment

for example. Coordinates are measure in x, y, z, each axis is in millimeters. Since the experiment is performed on a flat dish, x and y vary on the range of 0-100 (i.e., 10 centimeters), but the z axis only varies from 0-10 (i.e., a 1 cm high box). Normalizing such data with greatly emphasize the z axis, which most likely is not supported by a physical interpretation of the results. Last point, the goal of this project is to make prediction rather than analysis. That means it is not necessary to bring all the features to the same range to see the effect of features. Our goal is not to explain the effects of features. Overall, we decide not to perform normalization on our dataset.

Learning methods and algorithms

Let us assume we have a data set containing feature values and a vector of response as follows:

$$X = \begin{bmatrix} 1 & x_1^1 & \dots & x_p^1 \\ & \vdots & \vdots & \vdots \\ 1 & x_1^N & \dots & x_p^N \end{bmatrix}, Y = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix}.$$

1. Linear Regression

Given a data set $[X, Y]$, a linear regression model assumes that the relationship between the response value y and a vector of features x is linear. So, we assume that:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$

Thus, to obtain the model's parameters, we should solve the following optimization problem:

$$\min \|Y - \beta X\|^2 \ .$$

The output of our code for linear regression coefficients, using R is given below. The features which are specified by stars are the ones with lower importance because the model decided to assign them a small coefficient. (To see the coefficient of linear regression model, please see table 1 in the appendix).

2. Ridge Regression

Ridge regression tries to build a linear model to show the relationship between response and features. The difference between ridge regression and linear regression is that ridge regression tries to artificially prevent the model to have the best fit to the existing data. The formulation for this problem is as follows:

$$\min \frac{1}{N} \|Y - \beta X\|^2 + \lambda \|\beta\|_2^{\ 2} \ .$$

$\lambda \geq 0$ is not part of the model, it should be tuned before solving the formulation.

In our case, after training and hyperparameter tuning, we have that the model's parameters are as table 2 (please see the appendix) for $\lambda^* = 0.7905076$.

3. Least Absolute shrinkage and selection operator (LASSO)

Same as ridge regression, LASSO aims to build a linear model when it also adds a regularization expression to the objective function of linear regression. By adding this regularization expression, this model benefits from feature selection and avoiding from overfitting. The formulation for this problem is as follows:

$$\min \frac{1}{N}\|Y-\beta X\|^2+\lambda\|\beta\|_1{}^2 \ ,$$

Where $\lambda \geq 0$ is a hyperparameter.

Here, we have the summary of our model after training for $\lambda^* = 0.7905076$.

4. Elastic Net

Elastic Net is a combination of ridge regression and LASSO. The formulation for this method is given below:

$$\min \frac{1}{N}\|Y-\beta X\|^2+\lambda_1\|\beta\|_1{}^2 \ +\lambda_2\|\beta\|_2{}^2 \ ,$$

Where $\lambda_1, \lambda_2 \geq 0$ are hyperparameters and $\lambda_1 + \lambda_2 = 1$.

Here, we have the model's parameters which is tuned for $(\lambda_1{}^*, \lambda_2{}^*) = (0.06813, 0.93187)$. It is clear from the result that the coefficient associated with some features is zero, which means a feature section has occurred during training procedure. (More details are in Table 3 in the appendix.)

5. Support Vector Machine

Support Vector machine can also be used for regression problems. In contrast with linear regression, this method tries to build a linear model minimizing $L2$ norm of coefficient vector instead of squared error. In this method, we consider the violation from the linear function as constraints, and allowed violation is hyperparameter. So, the objective function and constraints are as follows:

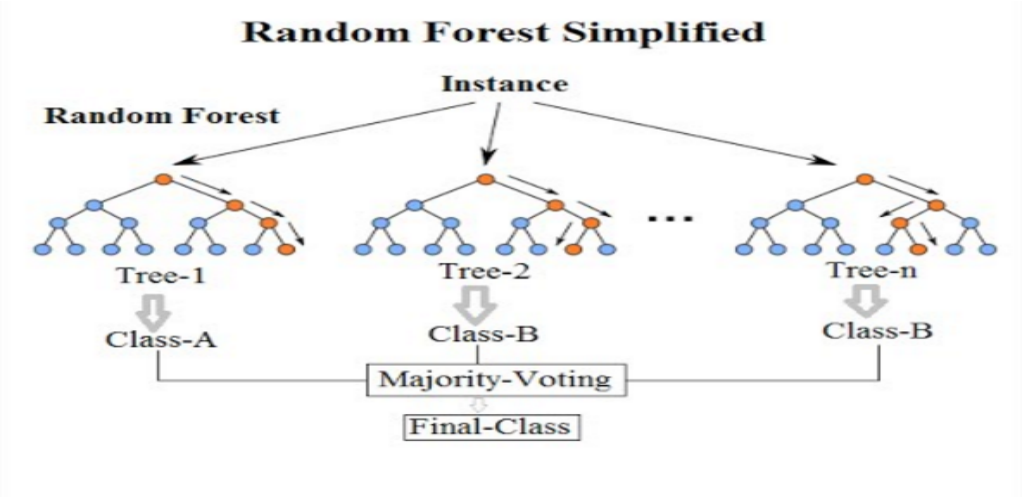
$$\min \frac{1}{2}\|W\| \ s.t: \left|y^i - wx^i\right| \leq \varepsilon$$

There are different options for kernel of SVM model in R. After trying polynomial kernel for $degree = 3$ and $gamma = (0.001, 0.01, 0.1, 1)$, and radial kernel for $gamma = (0.001, 0.01, 0.1, 1)$, we have the best model based on MSE is the one with the summary as presented below:

SVM-Type	SVM-Kernel	cost	gamma	epsilon	# of Support Vectors
eps-regression	radial	1	0.001	0.1	183

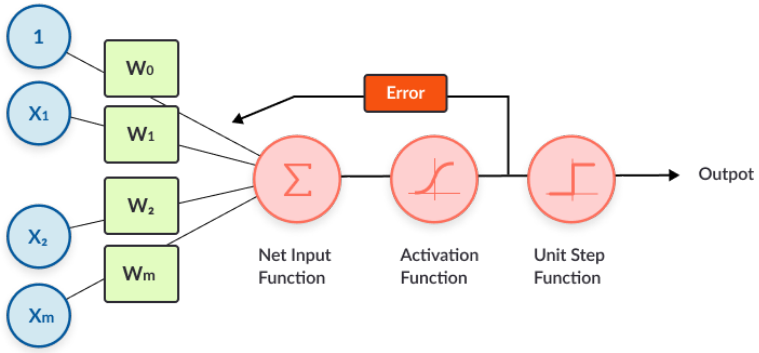
6. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.



7. Neural Network

Each neural network contains three different parts: input layer, output layer and processor . The features will be multiplied by some especial weights and then their summation is calculated. Finally, the summation will be amplified by an activation function.



Regression in neural networks

The error of training of a neural network from a given example is calculated by determining the difference between the processed output of the network and a target output . Then, the weights will be modified according to a learning process and error. By successive adjustments the neural network output will be increasingly similar to the target output. After a sufficient number of these adjustments the training can be terminated based on certain criteria.

Final results

Model comparison

We applied the statistical methods above to build models based on training set, aim to find the best model. We compared models with MSE using validation set. The model of lowest MSE means it has the best performance on predicting the response.

Table: Model Comparison (MSE)

Methods	MSE
Linear Regression	0.2166775
LASSO	0.1639627
Ridge Regression	0.1657413
Elastic Net	0.1642524
SVM	0.1862394
Random Forest	0.1840506
Neural Network	0.2166791

The table above shows the LASSO model has the best predicting ability. Therefore, we selected it as our final model.

Conclusions

We calculated test MSE using three penalized regression models, and the lowest test MSE was 0.08261.

Methods	MSE
LASSO	0.08261
Ridge Regression	0.10183
Elastic Net	0.09425

The LASSO model had the best predicting performance in testing set.

Appendix: Tables

Table 1

(Intercept)	**7.93E+00**	**3.613e+00 2.194**	**0.02939 ***
f1	1.15E-03	1.632e-03 0.707	0.48063
f2	-4.81E-02	7.845e-02 -0.613	0.54052
f3	4.50E-02	7.400e-02 0.609	0.54340
f4	-1.56E-01	8.498e-02 -1.830	0.06867 .
f5	1.44E-01	1.750e-01 0.825	0.41050
f6	3.46E-02	1.078e-01 0.321	0.74856
f7	-2.57E-01	8.771e-02 -2.927	0.00380 **
f8	-7.06E-02	1.178e-01 -0.599	0.54958
f9	-1.17E-01	4.805e-02 -2.430	0.01596 *
f10	5.36E-05	1.457e-04 0.368	0.71342
f11	-1.71E-04	3.736e-04 -0.459	0.64686
f12	9.01E-04	8.243e-03 0.109	0.91305
f13	-1.18E-02	1.852e-02 -0.635	0.52613
f14	5.93E-05	1.700e-04 0.349	0.72751
f15	2.72E-01	1.012e-01 2.685	0.00785 **
f16	-3.78E-03	2.219e-03 -1.703	0.09005 .
f17	-5.26E-03	1.102e-02 -0.477	0.63397
f18	8.37E-04	6.656e-04 1.258	0.20988
f19	-9.03E-04	2.387e-03 -0.378	0.70559

f20	9.34E-03	3.860e-03 2.421	0.01636 *
f21	1.27E-02	7.226e-03 1.752	0.08135 .
f22	-4.36E-01	7.028e-01 -0.620	0.53619
f23	-7.41E-03	6.210e-03 -1.192	0.23449
f24	-3.55E-03	3.006e-03 -1.181	0.23879
f25	9.81E-03	1.790e-02 0.548	0.58420
f26	1.74E-03	1.985e-02 0.087	0.93045
f27	1.46E-03	1.371e-03 1.068	0.28670
f28	-1.80E-03	9.308e-04 -1.933	0.05463 .
f29	1.47E+00	1.540e+00 0.953	0.34180
f30	-1.05E-02	1.096e-02 -0.958	0.33911
f31	4.31E-01	6.539e-01 0.659	0.51048
f32	-9.42E-03	1.423e-02 -0.662	0.50854
f33	-5.60E-02	2.120e-02 -2.640	0.00894 **
f34	6.99E-03	2.610e-03 2.676	0.00806 **
f35	4.47E-04	1.419e-02 0.031	0.97492
f36	1.21E-01	1.116e-01 1.082	0.28053
f37	5.78E-04	3.732e-03 0.155	0.87716

Table 2

intercept	5.679569
f1	0.000146183
f2	-0.001348461
f3	0.006287822
f4	0.004707072

f5	0.02734333
f6	0.001527991
f7	-0.008666436
f8	0.008257848
f9	-0.007794714
f10	7.66048E-06
f11	-1.09799E-05
f12	0.001017728
f13	-0.002609479
f14	1.01347E-05
f15	0.01032775
f16	2.70569E-05
f17	0.00048893
f18	0.000125409
f19	0.000235207
f20	0.000389168
f21	0.000998073
f22	0.0140615
f23	0.000489361
f24	0.000417019
f25	-0.001508431
f26	0.002724956
f27	0.0002334
f28	-0.000575897
f29	-0.09612169
f30	0.000389425
f31	-0.1910218
f32	4.48049E-05
f33	-0.00919167
f34	0.000642799
f35	0.000412971
f36	0.002101294
f37	0.000188919

Table 3

Intercept	**5.618381**
f1	0
f2	0
f3	0.01358297

f4	0
f5	0.07351039
f6	0
f7	-0.0303831
f8	0.00700949
f9	-0.0211565
f10	0
f11	0
f12	0
f13	0
f14	0
f15	0.05656663
f16	0
f17	0
f18	0.00018756
f19	0
f20	0.0011262
f21	0.00290597
f22	0
f23	0
f24	0
f25	-0.0022521
f26	0
f27	0.00077635
f28	-0.0015309
f29	0
f30	0
f31	-0.2200215
f32	0
f33	-0.0311738
f34	0.00177818
f35	0
f36	0
f37	0.00023168