STA1001 Data Analysis Project

Build Regression Models to Predict Computer Hardware Performance

LU GAN ecelu.gan@mail.utoronto.ca
Student number: 1003119608
YIFAN ZHANG ivana.zhang@mail.utoronto.ca
Student number: 1004245952

1. Introduction

In this project, we mainly use regression methods to build several models to predict computer hardware performance. Based on a source dataset, we get much information about performance value of different kinds of computer CPU and details of related attributes. We will firstly analysis the relationship among these attributes and the hardware performance. Then build a full model to include all attributes. With explanation about the validity of full model from overall aspects we have learned in class, we adjust it and build some other models with more regression algorithms. We will compare models and analysis their advantages and shortcomings.

The rest structure of this report is organized as follows. In the second section, we will introduce the source dataset including predictors and response variables in our project with some related statistics and distributions. We also give scatter matrix plots to show the relationship between pairs of variables. The third section will build first full model and make regression diagnostics analysis. to determine its validity from several aspects. In the next section, we make adjustments to select some other regression models and compare them. We summarize our results and conclusions in the last section. In the appendix, we attach all original code in this project.

2. Dataset

2.1 Data Source

We find the computer hardware dataset in the website of UCI Machine Learning Repository. Give the citation of the dataset.

Phillip Ein-Dor, Jacob Feldmesser, Faculty of Management, Tel Aviv University, Ramat-Aviv, Tel Aviv, Israel[1].

2.2 Data Exploration

Before analysis, we firstly need to import the dataset file and transform it to dataframe. The file only has numerical data so that we can add column names in read function. Figure 1 shows part of the dataframe.

We can observe that there are 10 columns in the dataframe except the first index column. The meanings of each column are vendor name (30 kinds, like Amdahl, IBM), model name (many unique symbols), machine cycle time in nanoseconds (MYCT), minimum main memory in kilobytes (MMIN), maximum main memory in kilobytes (MMAX), cache memory in kilobytes (CACH), minimum channels in units (CHMIN), maximum channels in units (CHMAX), published relative performance (PRP), estimated relative performance from the original article (ERP).

```
data=pd.read_csv('machine.data.txt',names=['Vendor','Model Name','MYCT','MMIN','MMAX','CACH',
                                           'CHMIN','CHMAX','PRP','ERP'])
data.head()
```

	Vendor	Model Name	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP	ERP
0	adviser	32/60	125	256	6000	256	16	128	198	199
1	amdahl	470v/7	29	8000	32000	32	8	32	269	253
2	amdahl	470v/7a	29	8000	32000	32	8	32	220	253
3	amdahl	470v/7b	29	8000	32000	32	8	32	172	253
4	amdahl	470v/7c	29	8000	16000	32	8	16	132	132

Figure 1

We can get type information and record number. The result is in Figure 2.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 10 columns):
Vendor        209 non-null object
Model Name    209 non-null object
MYCT          209 non-null int64
MMIN          209 non-null int64
MMAX          209 non-null int64
CACH          209 non-null int64
CHMIN         209 non-null int64
CHMAX         209 non-null int64
PRP           209 non-null int64
ERP           209 non-null int64
dtypes: int64(8), object(2)
memory usage: 16.4+ KB
```

Figure 2

In this project, we analyze the relationship between related attributes and computer hardware performance. We will build models with regression methods. So, we delete the first two columns whose type are string. Considering to predicting the accurate performance, we use values of the PRP column. Then we can drop the Vendor, Model Name and ERP columns. The resulted dataframe is shown in Figure 3.

```
df = data.drop(['Vendor','Model Name','ERP'], axis=1)
```

```
df.head()
```

	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
0	125	256	6000	256	16	128	198
1	29	8000	32000	32	8	32	269
2	29	8000	32000	32	8	32	220
3	29	8000	32000	32	8	32	172
4	29	8000	16000	32	8	16	132

Figure 3

We can also get many statistics of variables shown in Figure 4. The dataset includes 209 records totally without missing values.

```
df.describe()
```

	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
count	209.000000	209.000000	209.000000	209.000000	209.000000	209.000000	209.000000
mean	203.822967	2867.980861	11796.153110	25.205742	4.698565	18.267943	105.622010
std	260.262926	3878.742758	11726.564377	40.628722	6.816274	25.997318	160.830733
min	17.000000	64.000000	64.000000	0.000000	0.000000	0.000000	6.000000
25%	50.000000	768.000000	4000.000000	0.000000	1.000000	5.000000	27.000000
50%	110.000000	2000.000000	8000.000000	8.000000	2.000000	8.000000	50.000000
75%	225.000000	4000.000000	16000.000000	32.000000	6.000000	24.000000	113.000000
max	1500.000000	32000.000000	64000.000000	256.000000	52.000000	176.000000	1150.000000

Figure 4

From the plot of response variable distribution in Figure 5, we can observe that the PRP distribution is left skewed and most values lie in interval [0, 200].

```
sns.set_style('whitegrid')
sns.distplot(data['PRP'],bins=100,kde=False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8408e21b70>
```

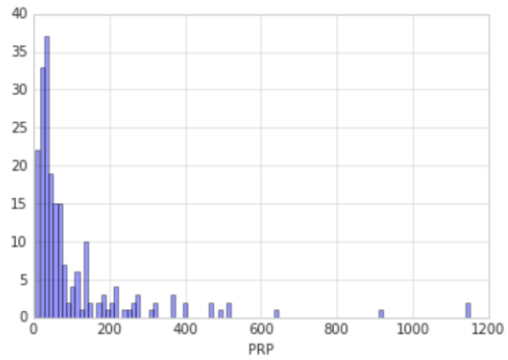

Figure 5

2.3 Scatter Matrix Plot

In this part, we plot the scatter matrix plot for both predictors and response variable. The plots are necessary and essential for us to analyze the relationship among the different predictors and the response variable. Pairs of the predictors appear to have a linear relation or at least linear approximately is an important evidence to determine the validity of models. For this question, the scatter matrix plot is shown in Figure 6. The plots in the diagonal are for variables themselves. From other plots, we can see that several predictors do not satisfy linear relationship like MMCT and others. This may lead that one of the conditions which can be used to judge how the model is misspecified not hold. If the plots of residual against predictors show pattern, we cannot get information about how the model is misspecified.

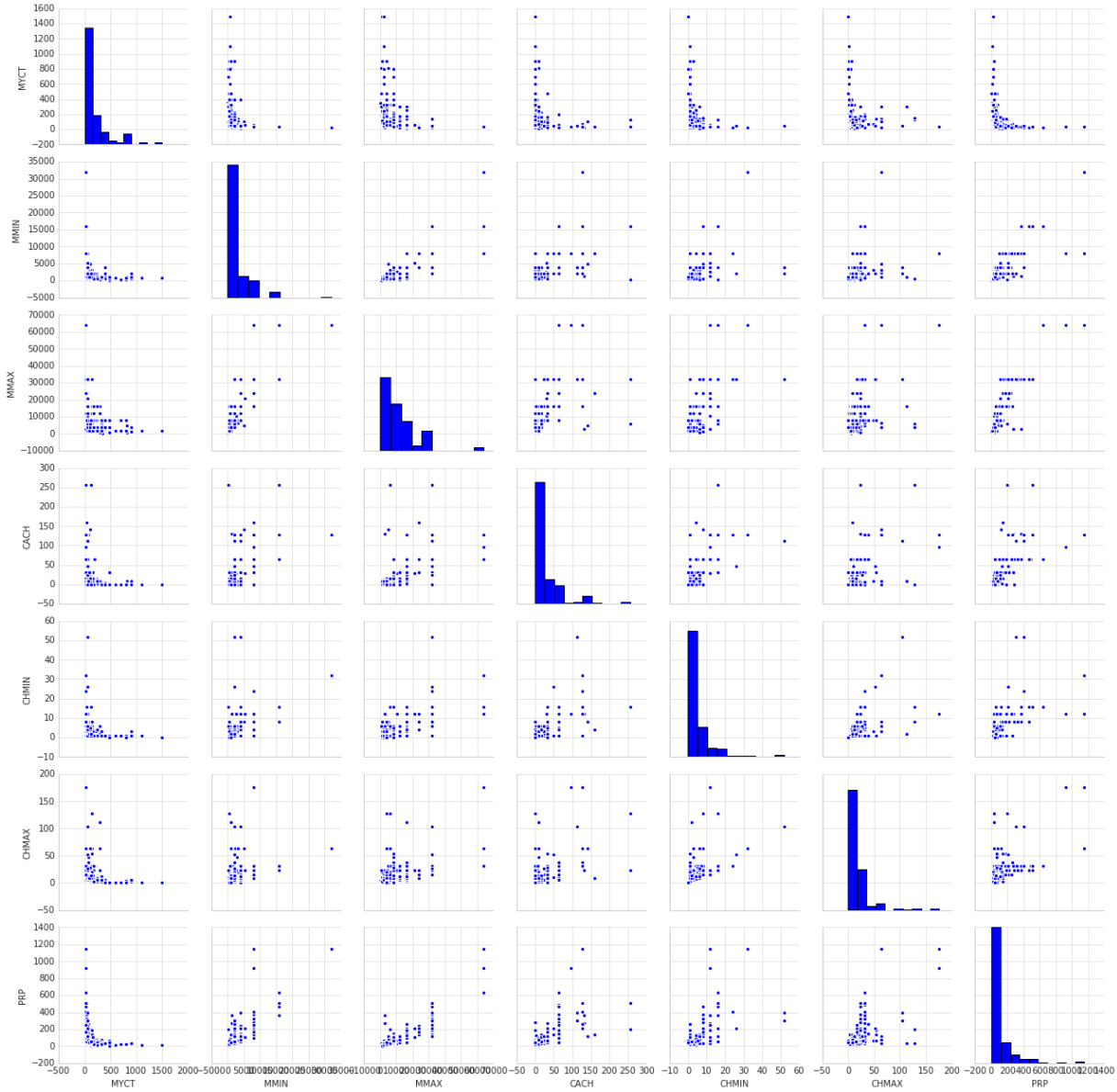

Figure 6

2.4 Correlation and Multicollinearity

We can use data analysis function to get a simpler visualization of correlations among variables in the data. The correlation matrix is shown in Figure 7. This plot provides related statistic about the multicollinearity exists in the data. When two. or more highly correlated predictors are included in a regression model, it could make the model be difficult to isolate an individual variable's influence on the response and be poorly estimated.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f44c4be3198>
```

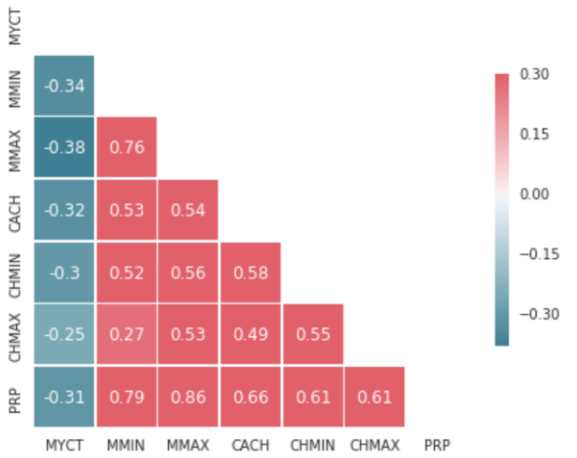

Figure 7

3. Model Selection with Regression Methods

3.1 Drop insignificant term

Firstly, we consider to building a full model with all predictors. Its form can be written as follows.

$$PRP = \beta_0 + \beta_1 MYCT + \beta_2 MMIN + \beta_3 MMAX + \beta_4 CACH + \beta_5 CHMIN + \beta_6 CHMAX + e$$

The model result is shown in Figure 8.

```
Call:
lm(formula = PRP ~ ., data = data_mod)

Residuals:
    Min      1Q  Median      3Q     Max
-195.82  -25.17    5.40   26.52  385.75

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.589e+01  8.045e+00  -6.948 5.00e-11 ***
MYCT         4.885e-02  1.752e-02   2.789   0.0058 **
MMIN         1.529e-02  1.827e-03   8.371 9.42e-15 ***
MMAX         5.571e-03  6.418e-04   8.681 1.32e-15 ***
CACH         6.414e-01  1.396e-01   4.596 7.59e-06 ***
CHMIN       -2.704e-01  8.557e-01  -0.316   0.7524
CHMAX        1.482e+00  2.200e-01   6.737 1.65e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.99 on 202 degrees of freedom
Multiple R-squared:  0.8649,    Adjusted R-squared:  0.8609
F-statistic: 215.5 on 6 and 202 DF,  p-value: < 2.2e-16
```

Figure 8

Based on the output, p value can show the significance of the relationship between each predictor and the response variable. We can observe that there is only one insignificant term, CHMIN, whose p value is larger than 0.05. From another aspect, the overall F-statistic is less than 2.2e-16, so that we reject the null hypothesis that all coefficients are equal to 0 (intercept-only model). Thus, we only drop the insignificant term CHMIN and get the first reduced model. We will determine its validity in later section. The reduced model is:

$$PRP = \beta_0 + \beta_1 MYCT + \beta_2 MMIN + \beta_3 MMAX + \beta_4 CACH + \beta_6 CHMAX + e$$

3.2 Backward algorithm

Backward elimination starts with all potential predictor variables and delete the predictor with the largest p-value at each step until all variables have been deleted from the model or the information criterion increases. The result of backward AIC variable selection based on this model are shown below in Figure 9:

```
Start:  AIC=1718.24
PRP ~ MYCT + MMIN + MMAX + CACH + CHMIN + CHMAX

        Df Sum of Sq    RSS    AIC
- CHMIN  1       359 727279 1716.3
<none>               726920 1718.2
- MYCT   1     27985 754905 1724.1
- CACH   1     76009 802929 1737.0
- CHMAX  1    163347 890267 1758.6
- MMIN   1    252179 979099 1778.5
- MMAX   1    271177 998097 1782.5

Step:  AIC=1716.34
PRP ~ MYCT + MMIN + MMAX + CACH + CHMAX

        Df Sum of Sq    RSS    AIC
<none>               727279 1716.3
- MYCT   1     28343 755623 1722.3
- CACH   1     78715 805995 1735.8
- CHMAX  1    177114 904393 1759.9
- MMIN   1    258252 985531 1777.8
- MMAX   1    270856 998135 1780.5
```

Figure 9

The final variable it selected is: MYCT, CACH, CHMAX, MMIN and MMAX.

3.3 Forward algorithm

Forward selection starts with no potential predictor variables and adds the predictor such that the resulting model has the lowest value of an information criterion until all variables have been added to the model or the information criterion increases. The result of forward AIC variable selection based on this model are shown below in Figure 10:

```
Start:  AIC=2124.58
PRP ~ 1

        Df Sum of Sq     RSS    AIC
+ MMAX   1   4007072 1373165 1841.2
+ MMIN   1   3399857 1980380 1917.7
+ CACH   1   2362428 3017809 2005.7
+ CHMIN  1   1994794 3385443 2029.8
+ CHMAX  1   1970664 3409573 2031.2
+ MYCT   1    507411 4872827 2105.9
<none>               5380237 2124.6

Step:  AIC=1841.17
PRP ~ MMAX

        Df Sum of Sq     RSS    AIC
+ CACH   1    297905 1075260 1792.1
+ MMIN   1    250276 1122889 1801.1
+ CHMAX  1    168097 1205068 1815.9
+ CHMIN  1    122926 1250239 1823.6
<none>               1373165 1841.2
+ MYCT   1      2413 1370752 1842.8

Step:  AIC=1792.06
PRP ~ MMAX + CACH

        Df Sum of Sq     RSS    AIC
+ MMIN   1    147995  927264 1763.1
+ CHMAX  1     70443 1004816 1779.9
+ CHMIN  1     20515 1054744 1790.0
+ MYCT   1     17613 1057647 1790.6
<none>               1075260 1792.1


Step:  AIC=1763.11
PRP ~ MMAX + CACH + MMIN

        Df Sum of Sq    RSS    AIC
+ CHMAX  1    171641 755623 1722.3
+ MYCT   1     22871 904393 1759.9
+ CHMIN  1     12094 915171 1762.4
<none>              927264 1763.1

Step:  AIC=1722.33
PRP ~ MMAX + CACH + MMIN + CHMAX

        Df Sum of Sq    RSS    AIC
+ MYCT   1   28343.4 727279 1716.3
<none>              755623 1722.3
+ CHMIN  1     717.5 754905 1724.1

Step:  AIC=1716.34
PRP ~ MMAX + CACH + MMIN + CHMAX + MYCT

        Df Sum of Sq    RSS    AIC
<none>              727279 1716.3
+ CHMIN  1    359.25 726920 1718.2
```

Figure 10

The final variable it selected is: MYCT, CACH, CHMAX, MMIN and MMAX.

3.4 Model comparison

From above methods, we come up with a same model that concluding five variables as MYCT, CACH, CHMAX, MMIN and MMAX. Now we compare this selected model with full model:

```
Analysis of Variance Table

Model 1: PRP ~ MMAX + MMIN + CHMAX + MYCT + CACH
Model 2: PRP ~ MYCT + MMIN + MMAX + CACH + CHMIN + CHMAX
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    203 727279
2    202 726920  1    359.25 0.0998 0.7524
```

Figure 11

From Figure 11, the p-value for the partial F-test is 0.7524, which is big enough to support the H0 hypothesis. Therefore, we delete the CHMIN term and adopt the new model.

4. Regression Diagnostics

In this chapter, we consider several regression diagnostics to check whether the model could satisfy basic assumptions. We use the reduced model we select above, containing terms MYCT, MMIN, MMAX, CACH and CHMAX. The R statistic output about the reduced model is shown in Figure 12. At this time, the p values of all predictors are significant and F-statistic is less indicating the model is valid and fitted better.

```
Call:
lm(formula = PRP ~ MMAX + MMIN + CHMAX + MYCT + CACH, data = data_mod)

Residuals:
    Min      1Q  Median      3Q     Max
-193.37  -24.95    5.76   26.64  389.66

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.608e+01  8.007e+00  -7.003 3.59e-11 ***
MMAX         5.562e-03  6.396e-04   8.695 1.18e-15 ***
MMIN         1.518e-02  1.788e-03   8.490 4.34e-15 ***
CHMAX        1.460e+00  2.076e-01   7.031 3.06e-11 ***
MYCT         4.911e-02  1.746e-02   2.813   0.0054 **
CACH         6.298e-01  1.344e-01   4.687 5.07e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.86 on 203 degrees of freedom
Multiple R-squared:  0.8648,    Adjusted R-squared:  0.8615
F-statistic: 259.7 on 5 and 203 DF,  p-value: < 2.2e-16
```

Figure 12

4.1 Leverage Point and Outliers

In order to analysis this dataset better, we take a look at leverage points. In this course, we have learned that leverage points have an unusually considerable effect on the estimated regression model, which could be estimated by Hat-Values. According to a popular rule [2], a data point will be regarded as a point of high leverage with p predictors if:

$$h_{ii} > 2 \times average(h_{ii}) = 2 \times \frac{p+1}{n}$$

In this data set, p = 6 and n = 209, so when $h_{ii} > 0.067$, the ith data point is of high leverage.

Also, outliers are the points which do not follow the pattern set by the bulk of the data, when one is taken into account the given model. We can evaluate from the standardized residual value. When the standardized residual of the points falls outside the interval from -2 to 2, this point could be considered as an outliner. We know that the ith standardized residual is given by:

$$r_i = \frac{\hat{e}_i - 0}{S\sqrt{1-h_{ii}}}, where \quad S = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\hat{e}_i^2}$$

We extract three outliers in this dataset and plot the graph Influence Plot in Figure 14 which is Standardized Residuals against Hat-values. It's clearly that the points both have higher Hat-Values

and outsider the range of standardized residuals are of bad leverage. The radius of circle indicates the influence of the data. Larger the radius, higher the influence. It's easy to find out cases 199, 9, 31 are of bad leverage, which will drag the regression line to the opposite direction and cannot tell the general pattern of the dataset. Therefore, we decide to drop those three points.

```
     rstudent unadjusted p-value Bonferonni p
200  8.606198         2.1226e-15   4.4363e-13
32   4.968796         1.4330e-06   2.9951e-04
10   3.841866         1.6345e-04   3.4161e-02

       StudRes        Hat     CookD
1    -2.806445 0.30810084 0.5653859
10    3.841866 0.32242582 1.0962840
32    4.968796 0.07387411 0.2939267
200   8.606198 0.22185198 2.5879425
```

Figure 13

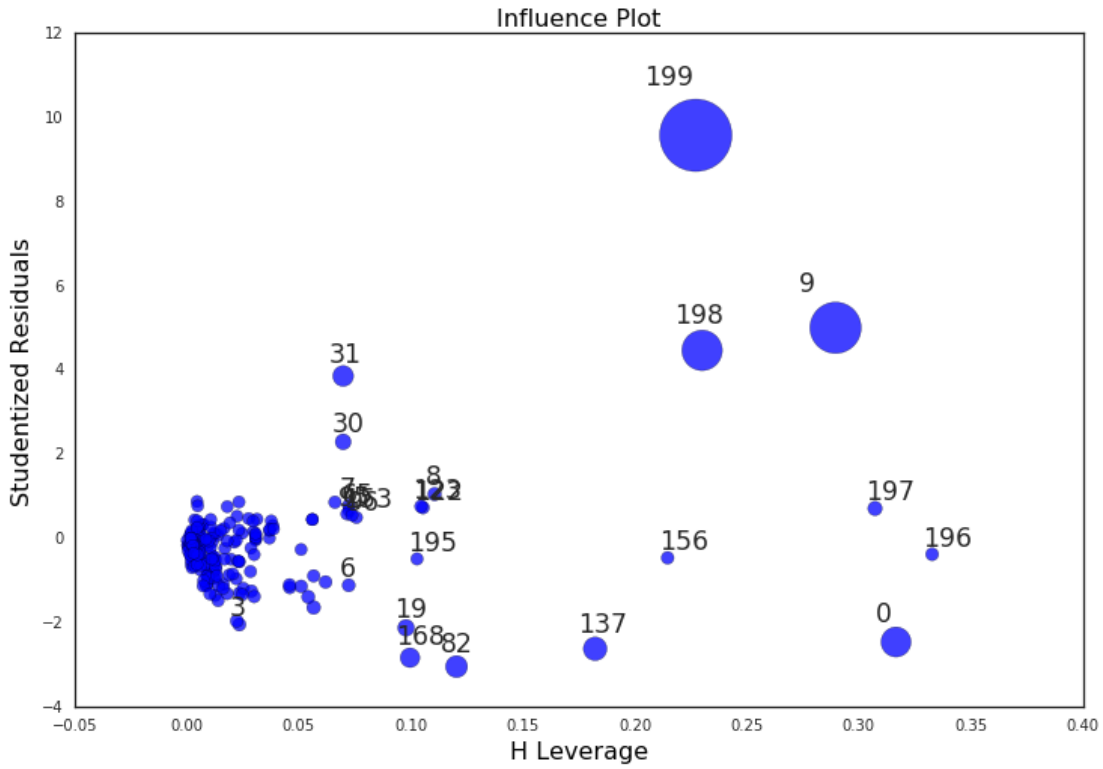

Figure 14

4.2 Standard residual

In multiple regression, when a valid model has been fit, a plot of standardized residuals, r_i against any predictor or any linear combination of the predictors (such as the fitted values) will have the following features [2]:

i. A random scatter of points around the horizontal axis, since the mean function of the e_i is zero when a correct model has been fit.

ii. Constant variability as we look along the horizontal axis.

We plot the graph that standardized residuals against each predictor as following in Figure 15. The

plots show a good performance regarding to the first feature. All points are roughly and evenly distributed on both sides of the horizontal line. We conclude that the variance of e is 0. However, the points distribute assembly and it's difficult to determine whether exists a pattern.

The plot in Figure 16 is PRP against the fitted values. The straight line fit to this plot provide a reasonable fit. This plot makes us more confidence in our model.

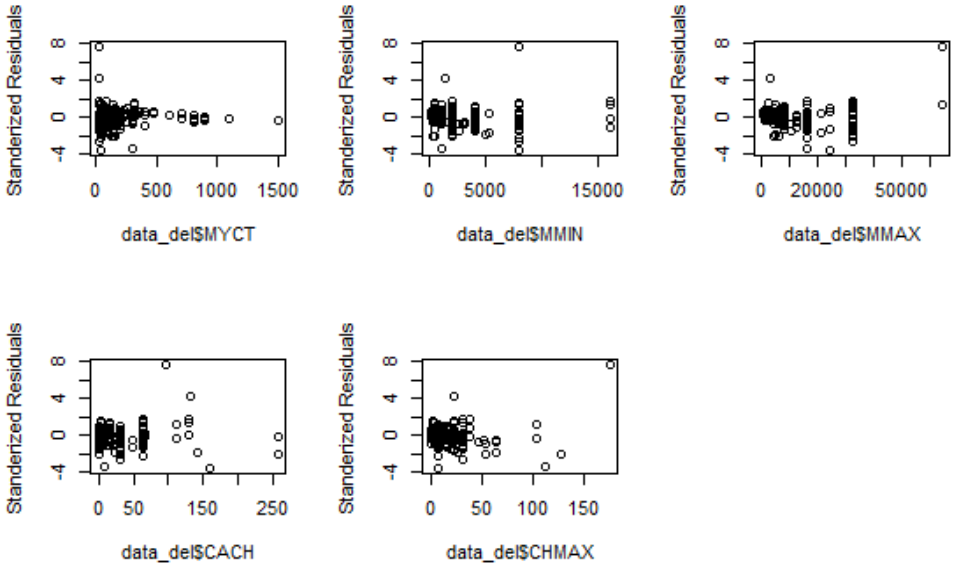

Figure 15

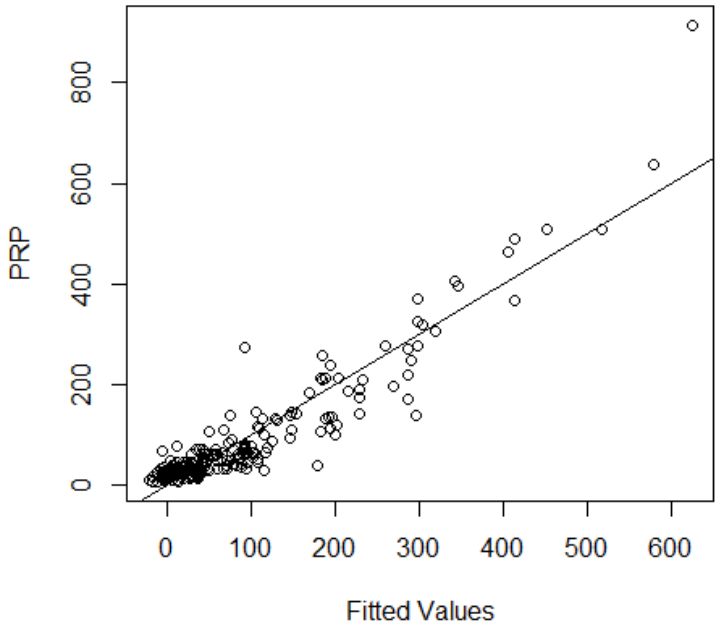

Figure 16

4.3 Added Variable Plot

We also plot added variable plots in Figure 17 and find out that in each plot, point 199th is really important to regression lines, especially in graph MMAX and CHMAX. This CPU is provided by vender Sperry and achieve 915 on published relative performance with large memory and quick speed.

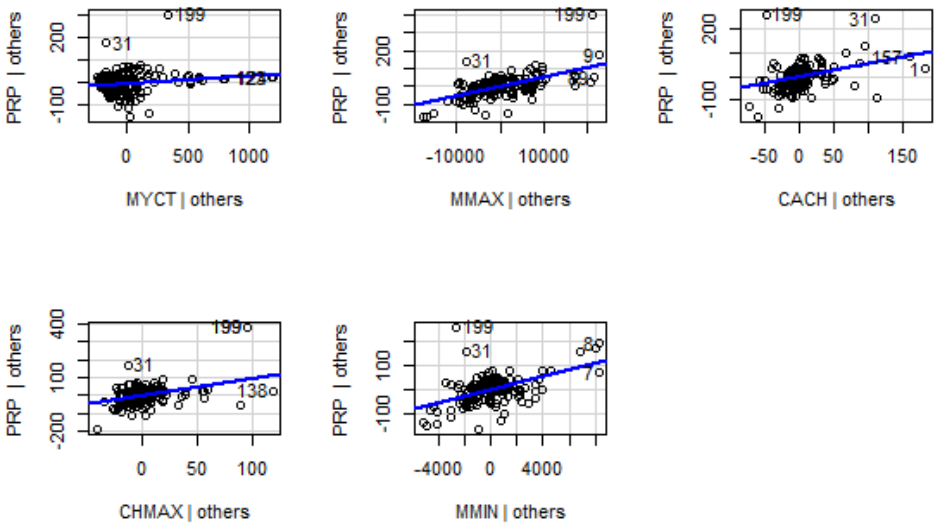

Figure 17

4.4 Diagnostic plot

iii. The top left plot in Figure 18 shows that residuals have non-linear patterns. There could be a non-linear relationship between predictor variables and PRP and the pattern could show up in this plot if the model doesn't capture the non-linear relationship. Therefore, this plot could also be an evidence to support that the model is valid [3].

iv. As for Q-Q plot, it indicates residuals are normally distributed when residuals are lined well on the straight dashed line.

v. We could find a horizontal line with equally (randomly) spread points in the left bottom graph Scale-Location plot. This plot shows if residuals are spread equally along the ranges of predictors and assumption of equal variance (homoscedasticity).

vi. The 199th point shows to be an extreme case against a regression line and can alter the results if we exclude them from analysis. This data point locates on the right top of the graph, which indicates that it is a bad leverage point.

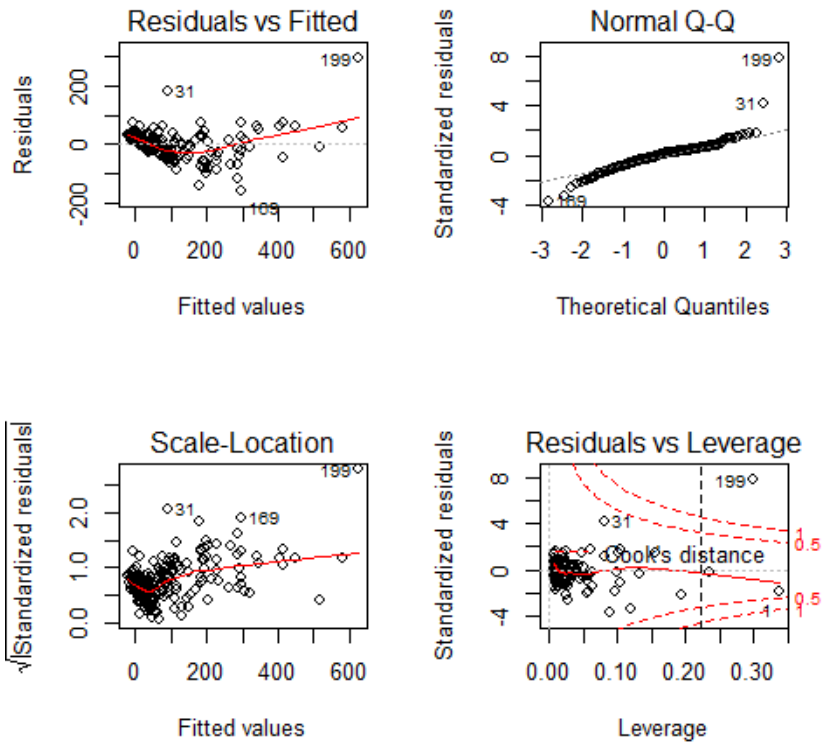

Figure 18

4.5 VIF

We also calculate the variance inflation factor for each predictor shown in Figure 19. All the variance inflation factors are smaller than 5, so multicollinearity is only a minor issue.

```
    MMAX     MMIN    CHMAX     MYCT     CACH
2.824096 2.676393 1.508250 1.222696 1.697402
```

Figure 19

5. Summary

In this project, we base on a computer hardware dataset and use related attributes and variables to build several models with regression methods. Firstly, we confirm the qualified predictors and response variables in this question. We make plots to analyze their relationship and distribution. Then we build three models through different regression algorithms, p value, backward and forward. After analysis and comparison, we get a reduced model with dropping one predictor. Next, we determine its validity from overall aspects including significance statistics, leverage points and outliers, standardized residual and diagnostic plots to prove that the reduced model is a valid one. In the analysis process of this project, we use both Python and R language. We mainly use Python

to format data and make the scatter plot matrix while R is applied to determine validity of the model.

6 Reference

[1] UCI Machine Learning Repository. "Computer Hardware Performance" [Online] Available: https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

[2] Sheather, S. "A Modern Approach to Regression with R." Springer, 2009

[3] Pydata, "linear regression in python". [Online] Available: http://songhuiming.github.io/pages/2016/12/31/linear-regression-in-python-chapter-2/. [Accessed: 20-June- 2018].

Appendix

A. R code

```
#import dataset
data <- read.table("machine.data.txt", header=TRUE, sep=",")
dim(data)

#delete unrelated columns
data_mod <- data[,c(3,4,5,6,7,8,9)]

dim(data_mod)
str(data_mod)
summary(data_mod)

#frequency histograms
par(mfrow=c(1,1))
p1 <- hist(data_mod$PRP,main="Computer Hardware Performance",xlab="published relative performance",ylab="Count",col=rgb(0,0,1,1/2))


#scatter plot matrix of predictor variables
pairs(data_mod)
pairs(~MYCT+MMIN+MMAX+CACH+CHMAX,data= data_mod,gap=0.4, cex.labels=1.5)

#correlation matrix
cor(data_mod)

attach(data_mod)

#variable selection
#backwards
m1 <- lm(PRP~., data=data_mod)
Backward_AIC <- step(m1,direction = "backward",data= data_mod)

#forwards
m2 <- lm(PRP~1, data=data_mod)
#n <- length(m2$residuals)
Forward_AIC <- step(m2, scope =list(lower=~1, upper=~MYCT+MMIN+MMAX+CACH+CHMIN+CHMAX), direction="forward", data=data_mod)

#select the model ERP ~ MMAX + MMIN + CHMAX + MYCT + CACH
m3 <- lm(PRP ~ MMAX + MMIN + CHMAX + MYCT + CACH, data = data_mod)
summary(m3)
```

```
par(mfrow=c(2,2))
plot(m3)

anova(m3,m1)

#Outliers
car::outlierTest(m3)
par(mfrow=c(1,1))
car::qqPlot(m3,main="QQ Plot")

par(mfrow=c(2,3))
car::leveragePlot(m3,term.name=MMAX)
car::leveragePlot(m3,term.name=MMIN)
car::leveragePlot(m3,term.name=CHMAX)
car::leveragePlot(m3,term.name=MYCT)
car::leveragePlot(m3,term.name=CACH)

# Influence Plot
par(mfrow=c(1,1))
influencePlot(m3, main="Influence Plot", sub="Circle size is proportial to Cook's Distance" )

#drop bad leverage points
data_del<-data_mod[-c(200, 10, 32), ]

detach(data_mod)

attach(data_del)
m3 <- lm(PRP ~ MMAX + MMIN + CHMAX + MYCT + CACH, data = data_del)

#variance inflation factors
car::vif(m3)

#residual plots
stan_residual <- rstandard(m3)
par(mfrow=c(2,3))
plot(data_del$MYCT,stan_residual,ylab="Standerized Residuals")
plot(data_del$MMIN,stan_residual,ylab="Standerized Residuals")
plot(data_del$MMAX,stan_residual,ylab="Standerized Residuals")
plot(data_del$CACH,stan_residual,ylab="Standerized Residuals")
plot(data_del$CHMAX,stan_residual,ylab="Standerized Residuals")

#Added-variable plots
par(mfrow=c(2,3))
car::avPlot(m3,variable=MYCT,ask=FALSE, main="")
car::avPlot(m3,variable=MMAX,ask=FALSE, main="")
car::avPlot(m3,variable=CACH,ask=FALSE, main="")
car::avPlot(m3,variable=CHMAX,ask=FALSE, main="")
car::avPlot(m3,variable=MMIN,ask=FALSE, main="")

#fitted value
par(mfrow=c(1,1))
plot(m3$fitted.values,PRP,xlab="Fitted Values")
abline(lsfit(m3$fitted.values,PRP))

#diagnostic plot
par(mfrow=c(2,2))
plot(m3)
abline(v=2*8/72,lty=2)

detach(data_del)
```

B. Python Code

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
import math
import matplotlib.pyplot as plt
%matplotlib inline

data=pd.read_csv('machine.data.txt',names=['Vendor','Model
Name','MYCT','MMIN','MMAX','CACH',
                                                    'CHMIN','CHMAX','PRP','ERP'])
data.head()

sns.set_style('whitegrid')
sns.distplot(data['PRP'],bins=100,kde=False)

df = data.drop(['Vendor','Model Name','ERP'], axis=1)
df.head()
df.describe()

sns.pairplot(df)

sns.set(style="white")

# Compute the correlation matrix
corr = df.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(7, 7))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5},annot=True)


from sklearn import datasets
import statsmodels.api as sm

X = df[['MYCT','MMIN','MMAX','CACH','CHMIN','CHMAX']]
y = df['PRP']
# Note the difference in argument order
model = sm.OLS(y, X).fit()
predictions = model.predict(X) # make the predictions by the model

# Print out the statistics
print(model.summary())

fig, ax = plt.subplots(figsize=(12,8))
fig = sm.graphics.influence_plot(model, ax=ax, criterion="cooks")
```