Predicting SaaS Revenue Multiples

29 April 2022

Introduction

How to value a company is one of the greatest debates in the history of finance. One of the most common ways that investors value companies is using revenue multiples. The appropriate revenue multiple to apply to a subject company is obtained from comparable public companies or precedent transaction multiples. Some industries have higher multiples than others. One industry that enjoys some of the highest revenue multiples is software and in particular SaaS which stands for Software as a Service. Some notable SaaS companies include Zoom, Slack, and Dropbox, to name a few. We set out to predict what the revenue multiple of a SaaS company is given several financial metrics. Additionally, we want to find out what gives some SaaS companies higher revenue multiples than others. We are particularity interested in these data questions because we all plan to work as software engineers as graduating. We will build a linear regression model and a random forest regression model to explore these data questions.

About the data

We define a revenue multiple as the enterprise value of a company divided by the trailing 12 months' revenue of the company. If a company has a 4 billion dollar enterprise value and did 200 million in revenue over the last 12 months, then its multiple would be 20x. All of the companies that we analyzed are publicly traded and we were able to obtain their financial data through lawfully required SEC filings via Ycharts. We put together a dat a set of 90 companies on our own but is based on the SEG SaaS Index 2022 Annual Report. SEG / Software Equity Group is a financial anlayst firm that focuses on analyzing SaaS companies. The index includes about 100 companies but we removed a handful that had incomplete data. Our response variable is ev_ttm_multiple which stands for enterprise value / trailing twelve months multiple. Our first predictor is revenue_growth which is revenue growth % year over year. Our second predictor is ebitda_margin % which is a way to measure the profitability of a company. Our third predictor is sales_efficiency which measures how effective a company is at turning sales and marketing spend into revenue. Our last predictor is gross_margin which measures the amount of profit made for every dollar in revenue.

Linear Regression Model

First we wanted to see the correlation of all the predictors just so we could get a good grasp and what variables would and wouldn't work. Next, we wanted to see if all or only some variables correlated in predicting our response variable. So we used the regsubsets() function that will essentially perform the best subset selection by identifying the best model with a given number of predictors. So in this case we have 4 predictors in total so the function tests for the best 1 variable model all the one to the best 4 variable model. We saw that our best model was the one with all 4 predictors. To confirm this we then created a table to show the rsq, adjusted r squared, Cp, and BIC for all of the best models from 1 predictor to 4. And we then decided that we would use the model with all four predictors for linear regression.

Here's the equation for the our linear regression model:

ev_ttm_multiple= $\beta 0$ + revenue_growth * $\beta 1$ + sales_efficiency * $\beta 2$ + gross_margin * $\beta 3$ + ebitda_margin * $\beta 4$ + c

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.2
```

```
StockData <- read_excel("stockdatasetfinal.xlsx")
#StockData <- read_excel("stockdatasetfinal.xlsx", sheet = "data")
#View(StockData)
```

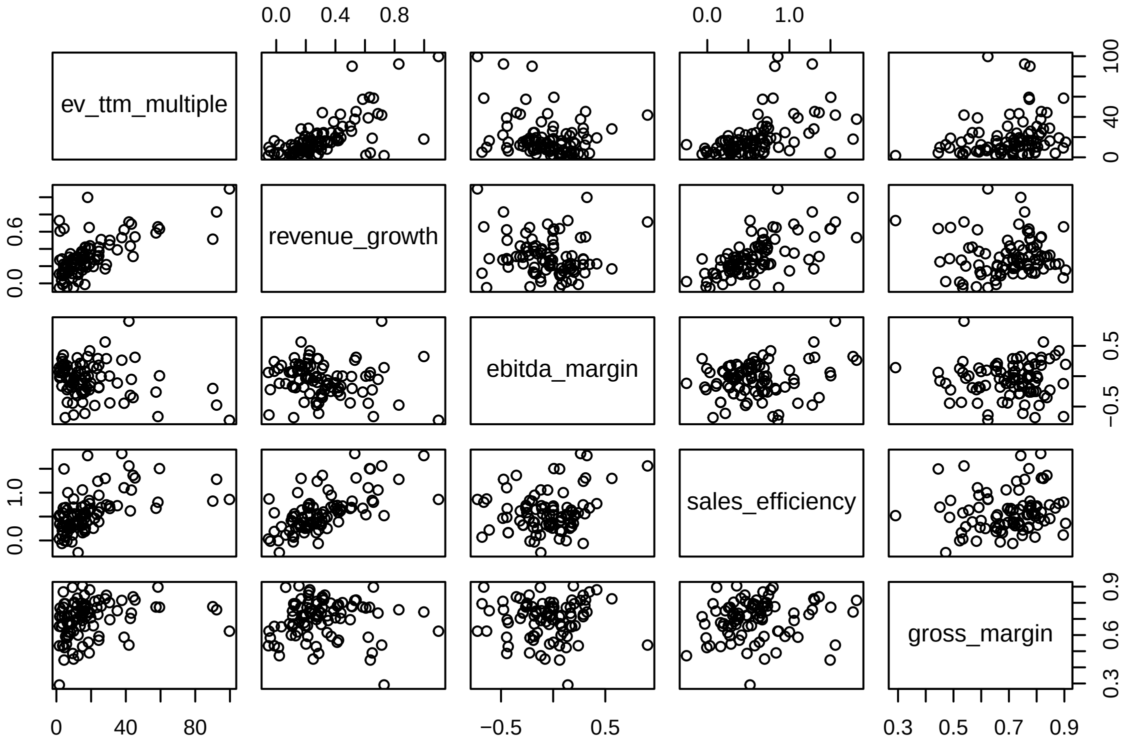
```
library(readxl)
StockData <-StockData[,-1] # remove the company names column
#View(StockData)
cor(StockData) # the correlation matrix
```

```
##                  ev_ttm_multiple revenue_growth ebitda_margin sales_efficiency
## ev_ttm_multiple        1.0000000    0.662793226   -0.24955498        0.5211693
## revenue_growth         0.6627932    1.000000000   -0.14141963        0.6429664
## ebitda_margin         -0.2495550   -0.141419625    1.00000000        0.2156177
## sales_efficiency       0.5211693    0.642966370    0.21561773        1.0000000
## gross_margin           0.2298569    0.009830959    0.01486597        0.1262192
##                  gross_margin
## ev_ttm_multiple   0.229856900
## revenue_growth    0.009830959
## ebitda_margin     0.014865968
## sales_efficiency  0.126219158
## gross_margin      1.000000000
```

While looking at the correlation matrix, we can see a good correlation between ev_ttm_multiple and revenue_growth, and a small correlation between ev_ttm_multiple and sales_efficiency.

Next we plot and check if there is any visible relation

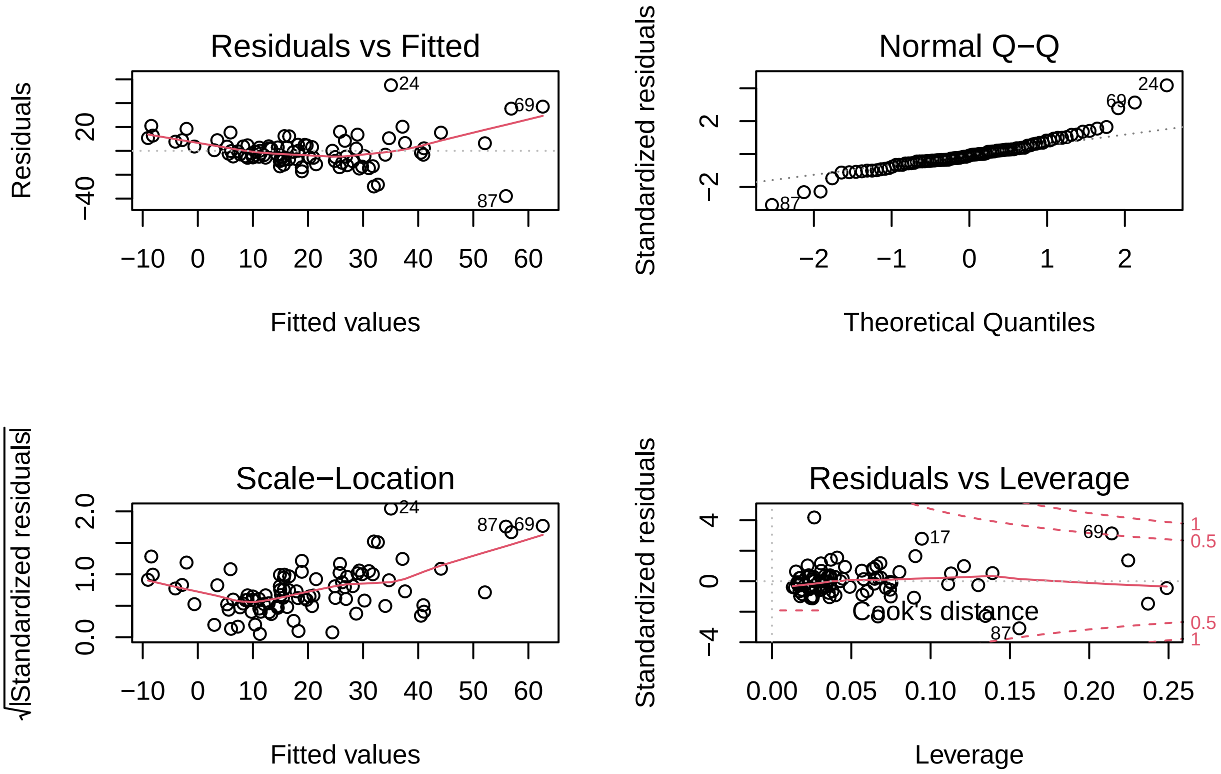
```
attach(StockData)
pairs(StockData)
```



We can notice a linear relation between ev__ttm__multiple and revenue__growth and sales__efficiency

Regsubstet told us that the best model was the one with all our models. this code performs a multiple linear regression. Notice the value of the mutiple R squared is about 54%.

```
model.lm <- lm(ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data=StockDa
par(mfrow=c(2,2))
plot(model.lm)
```



```
summary(model.lm)
```

```
##
## Call:
## lm(formula = ev_ttm_multiple ~ revenue_growth + sales_efficiency +
##     gross_margin + ebitda_margin, data = StockData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.928  -5.814  -1.195   4.924  55.042
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -23.672      8.771  -2.699  0.00839 **
## revenue_growth     40.315      8.926   4.516 2.01e-05 ***
## sales_efficiency   11.523      4.950   2.328  0.02228 *
## gross_margin       32.569     12.233   2.662  0.00928 **
## ebitda_margin     -16.314      5.518  -2.957  0.00402 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.35 on 85 degrees of freedom
## Multiple R-squared:  0.5442, Adjusted R-squared:  0.5227
## F-statistic: 25.37 on 4 and 85 DF,  p-value: 7.596e-14
```

The following code separate the data by training and test (cross validation) the training contain 80% of the data and the test contain 20%. Then we do a multiple linear regression.

```
set.seed(10)
sample<- sample.int(n=nrow(StockData), size=round(.80*nrow(StockData), 0), replace=F)
train<-StockData[sample, ]
test<-StockData[-sample, ]
model.lm <- lm(ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data=train)
```

This code does a prediction with the test and calculate the MSE.

```
model.pred<-predict(model.lm, newdata=test, se.fit = TRUE)
#mse.lm=mean((test$ev_ttm_multiple - model.pred$fit)^2)
mse.lm = mean((StockData$ev_ttm_multiple - predict(model.lm,StockData))[-sample]^2)
mse.lm
```

```
## [1] 144.9446
```

This code does Leave one out cross validation and calculate the MSE.

```
 library(boot)
 fit.glm = glm( ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data=StockDa
 cv.error.1=  cv.glm(StockData, fit.glm)$delta[1]
 cv.error.1
```

```
## [1] 205.1348
```

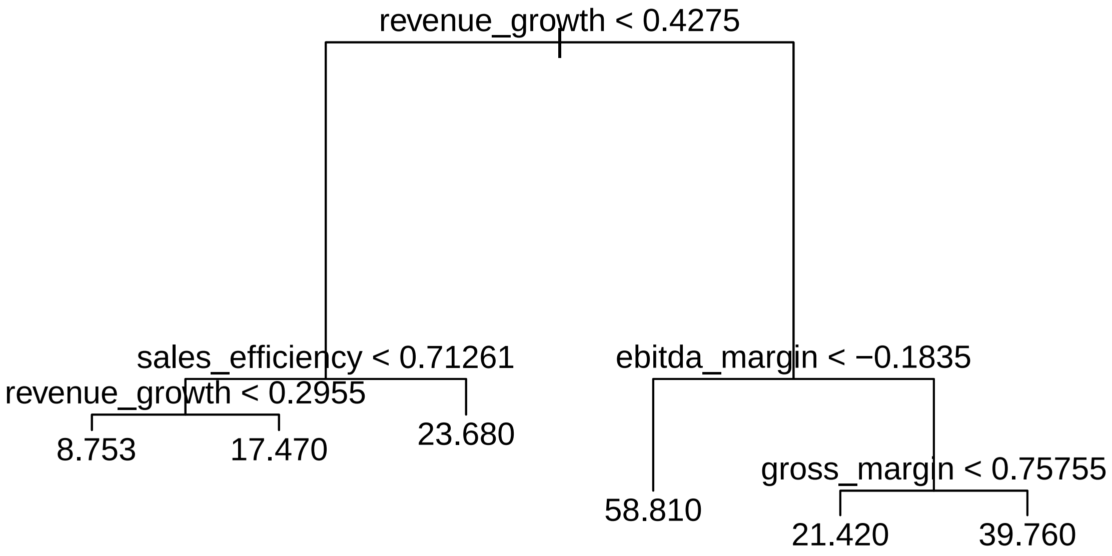
Random Forest Model

The reason we use this model is some of its advantages such as more accurate data, a more efficient way of handling data, and it also solves the issue of overfitting in a decision tree. We apply a tree-based model in our analysis plot(tree.model) which allows us to see the visualization of the data. Decision Trees are arranged in a hierarchical tree-like structure and are simple to understand and interpret. They are not susceptible to outliers and are able to capture nonlinear relationships.

```
set.seed(20)
library(tree)
tree.model <- tree(ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data=trai
summary(tree.model)
```

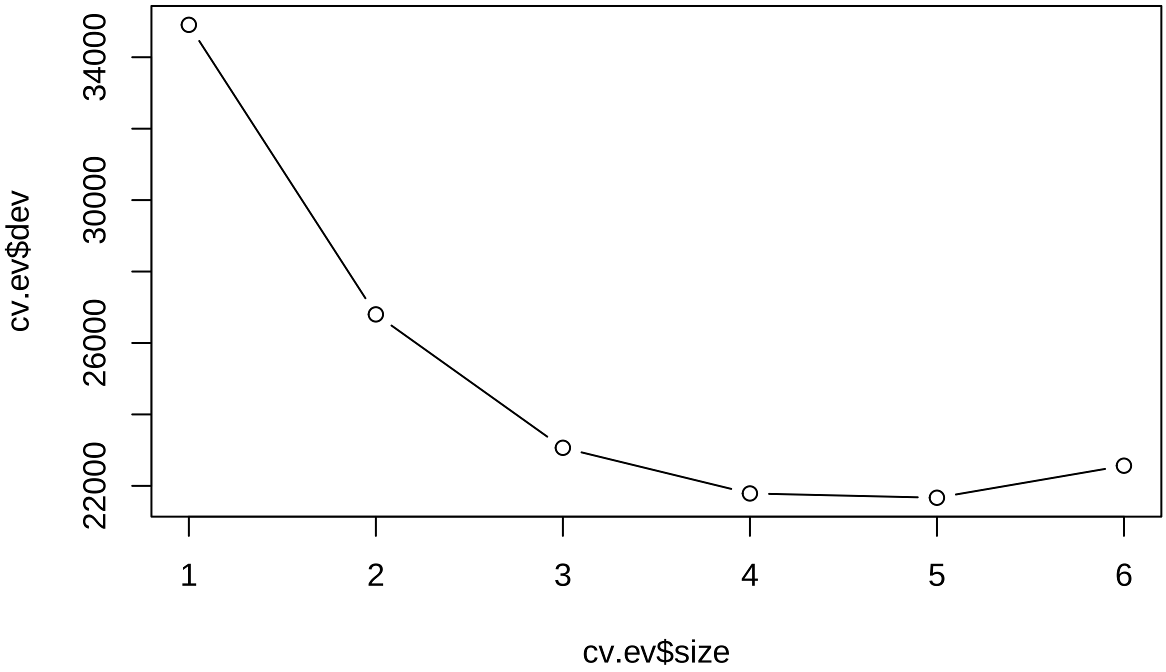
```
##
## Regression tree:
## tree(formula = ev_ttm_multiple ~ revenue_growth + sales_efficiency +
##     gross_margin + ebitda_margin, data = train)
## Number of terminal nodes:  6
## Residual mean deviance:  171.3 = 11310 / 66
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -39.910  -5.491  -1.012   0.000   4.990  40.890
```

```
plot(tree.model)
text(tree.model)
```

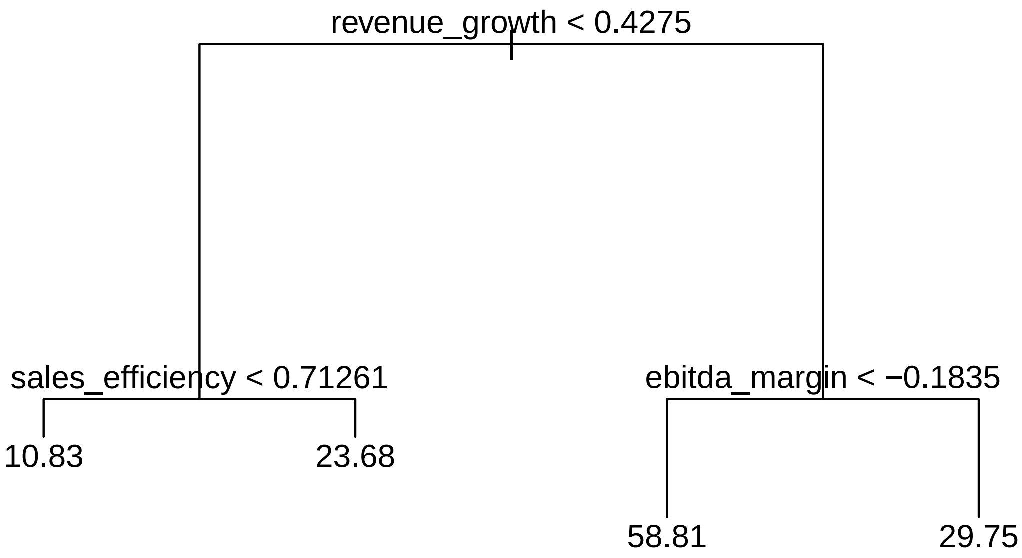


We also did our tree pruning model using the following code. Growing the tree beyond a certain level of complexity leads to overfitting. In our data, revenue_growth following sales_efficiency and the gross_margin don't have any impact on the target variable. Growing the tree beyond sales_efficiency is not going to add any value so we cut it there.

```
cv.ev=cv.tree(tree.model)
plot(cv.ev$size, cv.ev$dev, type="b")
```



```
pruned_tree = prune.tree(tree.model, best = 4)
plot(pruned_tree)
text(pruned_tree)
```



The following code does a cross validation on the tree model. This is our best model with the lowest MSE of 116.5

```
yhat.prune=predict(tree.model, newdata=test)
mse.tree=mean((test$ev_ttm_multiple - yhat.prune)^2)
mse.tree
```

```
## [1] 116.5161
```

```
# yhat.prune=predict(pruned_tree, newdata=test)
# mse.tree=mean((test$ev_ttm_multiple - yhat.prune)^2)
# mse.tree
```

The following code does a random forest: Notice the Mean squared residual and % of variability explained. In our model, we have 100 trees and we use random forest regression.

```
 p=4
B=100
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
bag.model=randomForest(ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data
bag.model
```

```
##
## Call:
##  randomForest(formula = ev_ttm_multiple ~ revenue_growth + sales_efficiency +      gross_margin + ebi
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 233.9423
##                     % Var explained: 36.68
```

The following code does boosting and shows the MSE.

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
set.seed(1)
boost.model = gbm(ev_ttm_multiple~revenue_growth+sales_efficiency+gross_margin+ebitda_margin, data=Stoc
train = sample(1:nrow(StockData), .5*nrow(StockData))
boost.test = StockData[-train, "ev_ttm_multiple"]
yhat.boost = predict(boost.model, newdata = StockData[-train,], n.trees = 100)
apply((yhat.boost-boost.test)^2, 2, mean)
```

```
## ev_ttm_multiple
##        177.5199
```

Conclusion

The random forest with pruning was our best model as it had the lowest MSE. However, the linear regression was able to give us more insight into which predictors are more associated with a higher revenue multiple. Revenue Growth ws our best predictor and sales efficiency was the second best. It is unreasonable to expect any model to predict the value of a company with high precision, however that are clear patterns that our models bring to light. It is logical that revenue growth plays a large role in the value of a SaaS company, considering that revenue is a company's oxygen. One of our most surprising finding was to see that ebitda margin % was somewhat negatively correlated with a higher multiple. A lower ebitda margin indicates that a company is "losing money," but it may be doing so to prioritize revenue growth which we found to be the most predictive. There is an expression in SaaS - grow at all costs and that may be true.

To tie everything back to why we chose to explore this data, every one of our group members plans to work professionaly in software after graduating. Stock is often big part of compensation packages. If any of us are considering a job offer from a SaaS company we will definitely be looking through their financials to see what kind of revenue multiple they should have.

Endnotes

Analysis for the linear regression model was performed by Oluwatobiloba Oladunjoye, Michael Thomas, and John Jackson. Analysis for the random forest regression model was performed by Mahamadou Dagnoko, Rafay Alam, and Min Shwe Maung Htet. To view our data set and source code, visit our GitHub repository https://github.com/johnjackson59/saas_multiples.