Model and Evaluation

The general problem description (based upon early submission)

What is the problem you want to solve?

The general problem is that software development companies might have a clear idea on the significance of different types of software architectural styles. Hence, the specific problem that companies need to understand the popularity of these software architectural styles.

Who is your client and why do they care about this problem? In other words, what will your client DO or DECIDE based on your analysis that they wouldn't have otherwise?

The potential client for this project could be software development companies, or research agencies. Based upon the data analysis, organiziations could gain a clear understanding on the usage of the different architechtural styles for software development, among various level of software developers.

-What data are you going to use for this? How will you acquire this data?

The data are available via Kaggle http://www.kaggle.com in a single dataset with various updated versions. These datasets are publicly available for downloading.

In brief, outline your approach to solving this problem (knowing that this might change later).

The variables of the project are software architechtural styles, software developers' job experiences, and their education levels. The data analysis will include linear regression and ranking, as well as relevant graph for data visualization.

What are your deliverables? Typically, this would include code, along with a paper and/or a slide deck.

The deliverables of the project include R code, a report (possibly be written by R markdown).

Data wrangling process

The following texts summarizes the important steps for cleaning up the data of software architechtural styles.

Renamed variables names to keep simple and meaningful

Some of the column names in the original dataset were relatively long. Short and simple names were given to represent the actual meaning of the columns. (See comments in R codes)

Separated variable "Industry"

The variable "Industry" contained various format of values taking from the original survey. Separating the responses to tidy the data.

Shortened variable "Education"

The variable "Education" contained redundant information. Removing the additional information to tidy the data.

Checked the uniqueness of observations

Some of the observations might be duplicated entries that have happened due to various reasons. Removing the duplicates when found to tidy the data.

Checked missing values

Some of the observations might be missing values that have happened due to various reasons. Omitting the observations that contain missing values when found to tidy the data (given the large amount of samples).

After the data wrangling step, a cleaned copy of dataset is ready for statistical analysis.

Preliminary findings

The following information shows the preliminary analysis to the software architechtural styles survey data.

First, read the cleaned data as a R object into the workspace and explore the structure of the data frame.

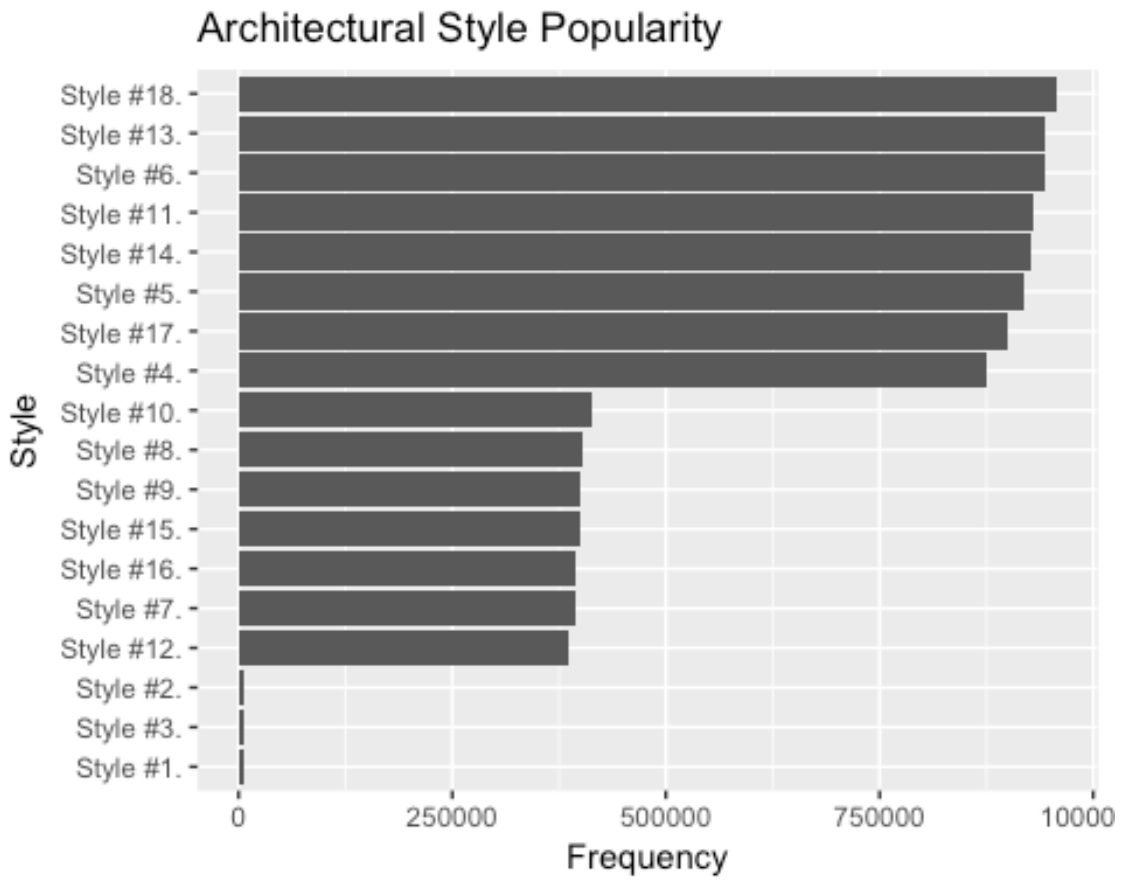
```
## 'data.frame':    1121 obs. of  23 variables:
##  $ Timestamp    : Factor w/ 999 levels "2016/12/11 1:41:33 PM
GMT+5",..: 47 49 50 53 54 60 61 61 62 2 ...
##  $ Name         : Factor w/ 669 levels " Inayat m","aalia",..: 356
390 171 660 290 26 300 300 153 621 ...
##  $ Organization.: Factor w/ 253 levels " KUST","Abbsen uni",..: 116
180 65 37 9 157 182 182 30 128 ...
##  $ Education    : chr  "BSC" "MS" "BSC" "BSC" ...
##  $ Industry     : chr  "Other" "Education" "Other" "Other" ...
##  $ Style #1.    : num  3 5 1 2 2 2 1 1 2 2 ...
##  $ Style #2.    : int  4 3 0 3 1 3 10 10 3 2 ...
##  $ Style #3.    : int  2 2 0 2 1 1 1 1 0 0 ...
##  $ Style #4.    : int  1455 50 0 1000 2 0 400 400 3 1000 ...
##  $ Style #5.    : int  2342 20 0 1500 1 6 15 15 100 50 ...
##  $ Style #6.    : num  1000 20 0 1700 0 18 10 10 100 30 ...
##  $ Style #7.    : int  859 10 0 500 3 6 5 5 100 20 ...
##  $ Style #8.    : num  456 20 0 600 0 7 10 10 15 0 ...
##  $ Style #9.    : num  232 5 0 600 1 2 5 5 0 1 ...
##  $ Style #10.   : num  965 10 0 500 4 1 0 0 2 1 ...
```

```
##  $ Style #11.   : num  2045 50 0 1500 3 ...
##  $ Style #12.   : num  543 10 0 400 1 1 50 50 2 0 ...
##  $ Style #13.   : int  2955 10 0 1600 1 6 300 300 5 0 ...
##  $ Style #14.   : int  1004 0 0 1400 1 13 1 1 500 2 ...
##  $ Style #15.   : num  356 0 0 300 0 0 0 0 100 0 ...
##  $ Style #16.   : int  289 10 0 500 0 0 10 10 3 0 ...
##  $ Style #17.   : num  2006 10 1 600 2 ...
##  $ Style #18.   : num  1594 20 0 1450 1 ...
```

A table to map styles

Style #1. Repository
Style #2. Client Server
Style #3. Abstract Machine
Style #4. Object Oriented
Style #5. Function Oriented
Style #6. Event Driven
Style #7. Layered
Style #8. Pipes Filters
Style #9. Data centeric
Style #10. Blackboard
Style #11. Rule Based
Style #12. Publish Subscribe
Style #13. Asynchronous Messaging
Style #14. Plug ins
Style #15. Micro kernel
Style #16. Peer to Peer
Style #17. Domain Driven
Style #18. Shared Nothing

Then, using a barplot to show the popularity of the different software architechtural styles.



These styles have actually been divided into three groups that are ranked by the popularity.

The most poupular styles are:

```
## [1] "Style #4."  "Style #5."  "Style #6."  "Style #11." "Style #13."
## [6] "Style #14." "Style #17." "Style #18."
```

Less poupular styles are:

```
## [1] "Style #7."  "Style #8."  "Style #9."  "Style #10." "Style #12."
## [6] "Style #15." "Style #16."
```

The least poupular styles are:

```
## [1] "Style #1." "Style #2." "Style #3."
```

Limitations

The dataset that has been utilized in the project might only contain the limited information about the software architechtual styles being used in the software development process. However, the dataset may demonstrate the popularity of the styles.

Updated approach

The next step going forward will be using the current data to build a logistic regression model (with 70% of the samples) as training data. Then, using the rest of 30% data to test the model, regarding the prediction on a particular software architectural style might be selected.

*More data wrangling steps have been identified to be necessary before the model creation. 1. Removed any unnecessary columns 2. Code Education variable 3. Code Industry variable

Building a logistic model to based on the 70% of dataset as the training data.

```
##
## Call:
## glm(formula = Industry ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4594  -0.8424  -0.7353   1.3869   1.9907
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.198e+00  2.658e-01  -4.506 6.61e-06 ***
## Education     2.864e-01  1.383e-01   2.071   0.0384 *
## `Style #1.`  -1.363e-02  4.637e-02  -0.294   0.7688
## `Style #2.`   1.692e-02  5.031e-02   0.336   0.7366
## `Style #3.`   1.232e-02  4.466e-02   0.276   0.7827
## `Style #4.`   5.280e-06  1.558e-04   0.034   0.9730
## `Style #5.`   9.647e-05  1.787e-04   0.540   0.5893
## `Style #6.`  -2.675e-05  1.660e-04  -0.161   0.8719
## `Style #7.`  -2.821e-04  3.857e-04  -0.731   0.4645
## `Style #8.`  -7.691e-04  4.597e-04  -1.673   0.0943 .
## `Style #9.`   3.780e-04  4.209e-04   0.898   0.3692
## `Style #10.` -7.661e-05  4.211e-04  -0.182   0.8557
## `Style #11.`  5.531e-06  1.508e-04   0.037   0.9707
## `Style #12.`  5.857e-04  4.227e-04   1.386   0.1659
## `Style #13.`  6.762e-05  1.555e-04   0.435   0.6636
## `Style #14.` -2.917e-04  1.677e-04  -1.739   0.0820 .
## `Style #15.` -4.084e-04  4.070e-04  -1.004   0.3156
## `Style #16.` -5.225e-04  4.067e-04  -1.284   0.1990
## `Style #17.`  2.080e-04  1.742e-04   1.194   0.2323
## `Style #18.`  1.433e-04  1.532e-04   0.935   0.3497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 935.06  on 784  degrees of freedom
## Residual deviance: 911.91  on 765  degrees of freedom
## AIC: 951.91
##
## Number of Fisher Scoring iterations: 4
```

From the summary of the model, few styles have the ability to predict whehter they can be used in the software industry.

```
##
##     FALSE TRUE
##   0   236    5
##   1    95    0
```

The model has the accuracy of

```
## [1] 0.7172619
```

The baseline method will get the accuracy at

```
## [1] 0.7172619
```

So, the model just beats the baseline method.

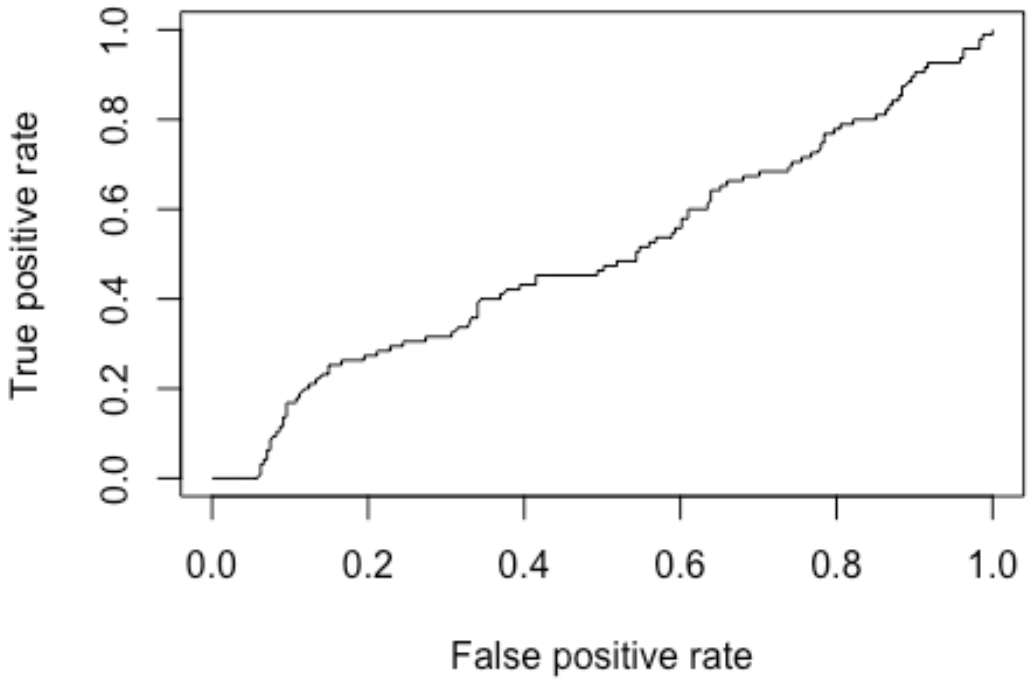
However, the following plot show the performance of the model.

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

## [1] 0.5015287
```



Recommedations

1. The accuracy of the model was very close to the baseline model, however, the further investigation should be used to improve the performance of the model.

2. The future research should focuse on a few styles that have the most popularity as the independent variablbes.

3. A different type of model may be used to explore the relationship between the groups of styles that have different levels of popularity, in order to achieve more powerful prediction.