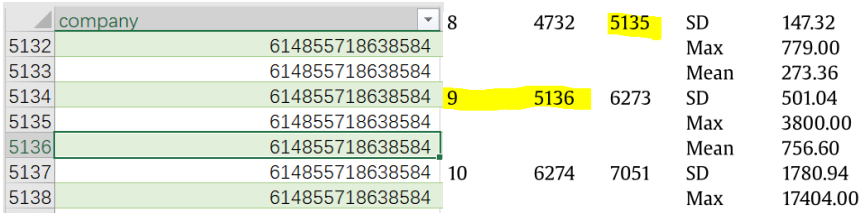
# How Could Company Increase the Engagement of Customers on Facebook?
## --Social Media Analytics about Thai Cosmetics

Live streaming commerce is booming in Thailand. Indeed, Thailand ranks first in the world in terms of the proportion of live streaming domestic viewers. Furthermore, Thailand has the highest proportion of consumers buying directly from social media in the world and is considered to be the most advanced market for e-commerce where people purchase goods from businesses freely and efficiently via instant messaging platforms.

As a result, social networking platforms have become a battleground for Thai companies. The more traffic a company's account can generate on social platforms, the better the promotion effect will be on its own products. With the growth of social media, people used Facebook more frequently than before. In 2016, Facebook introduced live streaming and emoji reactions as alternatives to traditional likes to add interest for users to the platform. At the same time, there has been a huge increase in the number of posts and engagements on company accounts. Companies can also use more Facebook metrics to collect data from their followers which were regarded as potential and current customers. In this report, I would make some suggestions on how to increase user's engagement with Facebook.

From the description paper, we could know that the dataset we used consists of 7050 Facebook posts of various types including status, link, videos, and photos. These posts were taken out from the Facebook page of 10 Thai fashion and cosmetics retail sellers from March 2012 to June 2018. For each Facebook post, the dataset records the resulting engagement metrics consisting of **shares, comments, and likes** and **emoji reactions** comprising **"love", "wow", "haha", "sad" and "angry"**. The emoji reactions which we distinguish traditional "likes" have been introduced after April 1st, 2016.

In this case, I use excel to conduct data preparation. I find some interesting points. First, the dataset only collected 9 companies' data not 10 as the paper said. Company 8 and 9 have the same post status id. Second, Only Company A have posts before Facebook introduced emoji reactions, which means other 8 companies came to Facebook after April 1st, 2016. Third, numbers of reactions equal to the sum of likes, loves, hahas, wows, angrys, sads.



| | company | | | | | |
|---|---|---|---|---|---|---|
| 5132 | 614855718638584 | 8 | 4732 | 5135 | SD | 147.32 |
| 5133 | 614855718638584 | | | | Max | 779.00 |
| 5134 | 614855718638584 | 9 | 5136 | 6273 | Mean | 273.36 |
| 5135 | 614855718638584 | | | | SD | 501.04 |
| 5136 | 614855718638584 | | | | Max | 3800.00 |
| 5137 | 614855718638584 | 10 | 6274 | 7051 | Mean | 756.60 |
| 5138 | 614855718638584 | | | | SD | 1780.94 |
| | | | | | Max | 17404.00 |

| Count:status_type Col | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Row-emoji-in | A | B | C | D | E | F | G | H | I | SUM |
| 0 | 1723 | | | | | | | | | 1723 |
| 1 | 912 | 1212 | 125 | 56 | 195 | 255 | 252 | 1542 | 778 | 5327 |
| **SUM** | **2635** | **1212** | **125** | **56** | **195** | **255** | **252** | **1542** | **778** | **7050** |

| num_reactions | sum | num_reaction-sum | num_likes | num_loves | num_wows | num_hahas | num_sads | num_angrys |
|---|---|---|---|---|---|---|---|---|
| 529 | 529 | 0 | 432 | 92 | 3 | 1 | 1 | 0 |
| 150 | 150 | 0 | 150 | 0 | 0 | 0 | 0 | 0 |

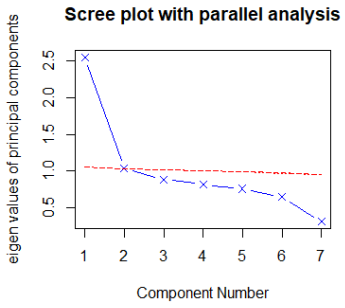
For the dataset, there is 15 original column including company, status_type, emoji_in, date, time, hour, num_reactions, num_shares, num_comments, num_likes, num_loves, num_wows, num_hahas, num_sads, num_angrys.

For the **company,** I use 9 letters A to I to represent 9 companies and for **emoji_in,** it is like a Boolean variable. If the post date is after April 1st, 2016(including April.1st 2016), it will be 1, and if the post date is before April 1st, 2016, it will be 0. And for the **hour,** it is separated from the time, it is a character to represent the 24 different hours for posts in a day.

| company | status_type | emoji_in | date | time | hour | num_reactio | num_commen | num_shares | num_likes | num_loves | num_wows | num_hahas | num_sads | num_angrys |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | video | 1 | 2018-04-22 | 6:00 | 6 | 529 | 512 | 262 | 432 | 92 | 3 | 1 | 1 | 0 |
| A | photo | 1 | 2018-04-21 | 22:45 | 22 | 150 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 |

I divided these 15variables into three parts. Date and time are characters about the timeline, not the factors. The 9 Facebook metrics reactions, shares, comments, likes, loves, wows, hahas, sads, angrys could be used to describe the engagement of posts' viewers. The 9 companies, 4 kinds of status type (photo, link, status, video), emoji_in and the 24 hours should be considered as features. I will use num_shares as dependent variable and status_type, emoji_in, hour as independent variables. Because according to Facebook metrics, shares would be the most powerful way to promote the content of the post at the same time, describe the engagement very well. Although comments have a big number than shares, they could not show on the viewers' page.

For the model building, First, I use R to conduct PCA of the other 7 metrics[1]: comments, likes, loves, wows, hahas, sads, angrys. I take 3 factors because the result is much better than using 2.




```
Principal Components Analysis
Call: principal(r = trainingSet[, c(8, 10:15)], nfactors = 3, rotate = "varimax",
    scores = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix
               RC1   RC2   RC3   h2   u2 com
num_comments 0.59  0.43 -0.05 0.53 0.47 1.8
num_likes    0.06  0.10  0.94 0.90 0.10 1.0
num_loves    0.85  0.17  0.19 0.78 0.22 1.2
num_wows     0.62 -0.22  0.40 0.59 0.41 2.0
num_hahas    0.73  0.01  0.04 0.54 0.46 1.0
num_sads     0.08  0.89  0.10 0.80 0.20 1.0
num_angrys   0.49  0.30 -0.05 0.33 0.67 1.7

Mean item complexity =  1.4
Test of the hypothesis that 3 components are sufficient.

The root mean square of the residuals (RMSR) is  0.12
 with the empirical chi square  2997.44  with prob <  0

Fit based upon off diagonal values = 0.81> round(pc$weights, 2)
```

Since I choose num_shares as the dependent variable, and status_type, emoji_in, hour as independent variables. I will use Linear Regression Model, SVM Model, and Random Forest to predict the features.

Linear Regression Model

```
fit1 <- lm(num_shares ~ .,
           data= trainingSet_IV_DV, na.action = na.exclude)
model.regression <- stepAIC(fit1, direction="backward")
testSet.predictRegression <- predict(model.regression, testSet_IV_DV)

fit2 <- lm(num_shares ~ .  + emoji_in:status_type_video
           + emoji_in:status_type_link + emoji_in:status_type_status,
           data= trainingSet_IV_DV, na.action = na.exclude)
model.regression2 <- stepAIC(fit2, direction="backward")
testSet.predictRegression2 <- predict(model.regression2, testSet_IV_DV)
```

```
> R2_Score(testSet.predictRegression, testSet$num_shares)
[1] 0.1266476
> RMSE(testSet.predictRegression, testSet$num_shares)
[1] 82.05665
> MAE(testSet.predictRegression, testSet$num_shares)
[1] 51.76357
> R2_Score(testSet.predictRegression2, testSet$num_shares)
[1] 0.1734352
> RMSE(testSet.predictRegression2, testSet$num_shares)
[1] 79.82841
> MAE(testSet.predictRegression2, testSet$num_shares)
[1] 45.81644
```

[1] Since reaction=likes+loves+hahas+sads+wows+angrys, I remove it.

I create 2 linear regression models. Since there will be some interaction between independent variables. For example, status type and emoji_in. So I add the cross variable including: emoji_in:status_type_video, emoji_in:status_type_link,emoji_in: status_type_status. It will increase the fitting of the model. But I do not use all the cross variables due to the processing time of the R program. Measures of fit2 will be better than fit1 according to the R square RMSE and MAE.

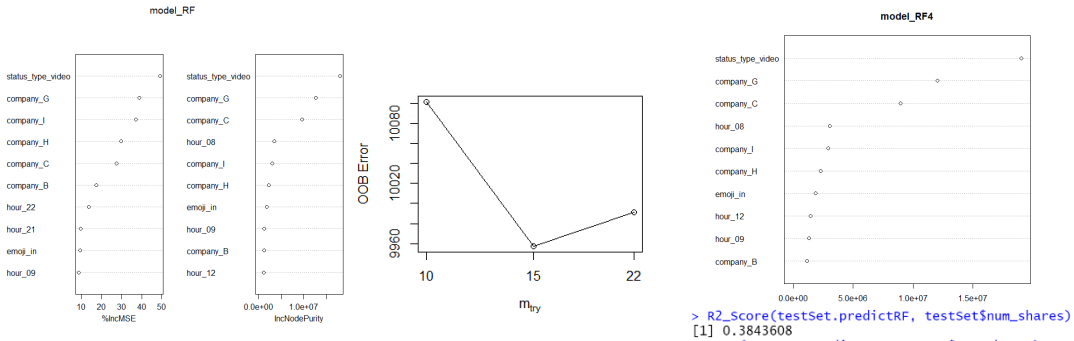
SVM Model

```
model_svm <- svm(trainingSet_IV, trainingSet$num_shares, kernel="linear")
coef(model_svm)
testSet.predictSVM <- predict(model_svm, testSet_IV)
model_svm2 <- svm(trainingSet_IV, trainingSet$num_shares,
                  kernel="polynomial", degree="2")
testSet.predictSVM2 <- predict(model_svm2, testSet_IV)
```

```
> # SVM 1
> R2_Score(testSet.predictSVM, testSet$num_shares)
[1] 0.2068706
> RMSE(testSet.predictSVM, testSet$num_shares)
[1] 78.19717
> MAE(testSet.predictSVM, testSet$num_shares)
[1] 36.06248
> # SVM 2
> R2_Score(testSet.predictSVM2, testSet$num_shares)
[1] 0.4005686
> RMSE(testSet.predictSVM2, testSet$num_shares)
[1] 67.98119
> MAE(testSet.predictSVM2, testSet$num_shares)
[1] 32.12792
```

I also create 2 SVM models by adding the square variables to SVM model2. And the fitting will be better than Linear Regression Model. But from the SVM model we could not get the significance of each independent variable.

Random Forest




```
model_RF <- randomForest(trainingSet_IV, trainingSet$num_shares,
                         ntree=100, mtry=15, importance=TRUE)
model_RF4 <- tuneRF(trainingSet_IV, trainingSet$num_shares,
                    15, ntreeTry=100, stepFactor=1.5,
                    improve=0.05, trace=TRUE, plot=TRUE, doBest=TRUE)
```

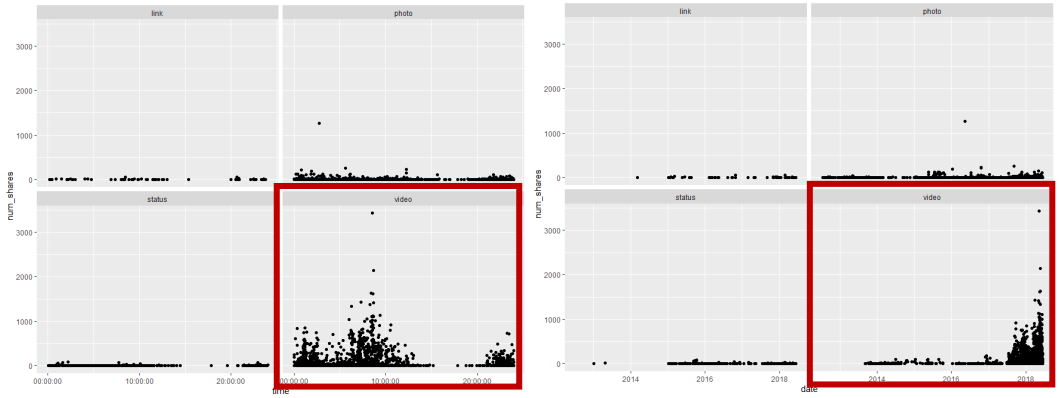
```
> R2_Score(testSet.predictRF, testSet$num_shares)
[1] 0.3843608
> RMSE(testSet.predictRF, testSet$num_shares)
[1] 68.89412
> MAE(testSet.predictRF, testSet$num_shares)
[1] 26.25745
> R2_Score(testSet.predictRF4, testSet$num_shares)
[1] 0.4137285
> RMSE(testSet.predictRF4, testSet$num_shares)
[1] 67.23082
> MAE(testSet.predictRF4, testSet$num_shares)
[1] 28.60189
```

It seems the Random Forest process the best result. And I try to tune the Random Forest model to RF4 and it returns a better result. From this model, the video type of post and several companies like company G, C, I, and specific hours like hour_8 and hour_9, will have significant effects on shares.

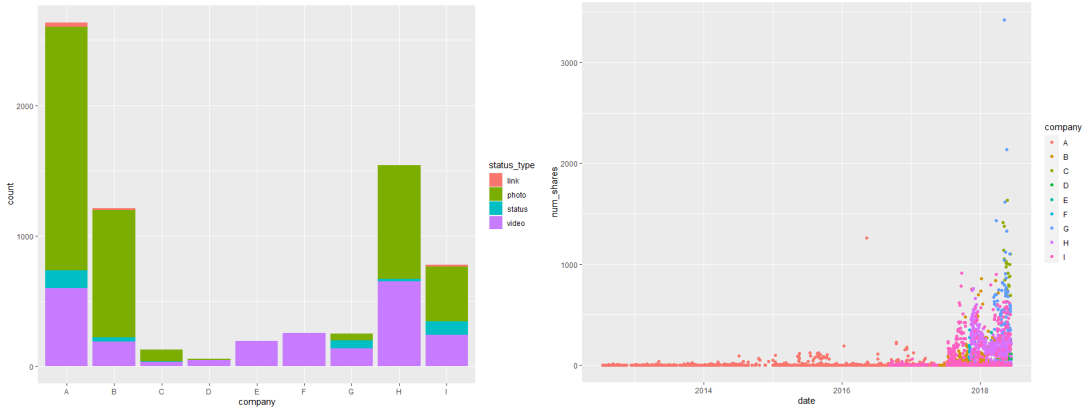
Business Insight

Video contributes a lot to the engagement of viewers. The table below shows that from 4 types of status, the video has a significant influence on the numbers of shares. For cosmetics and

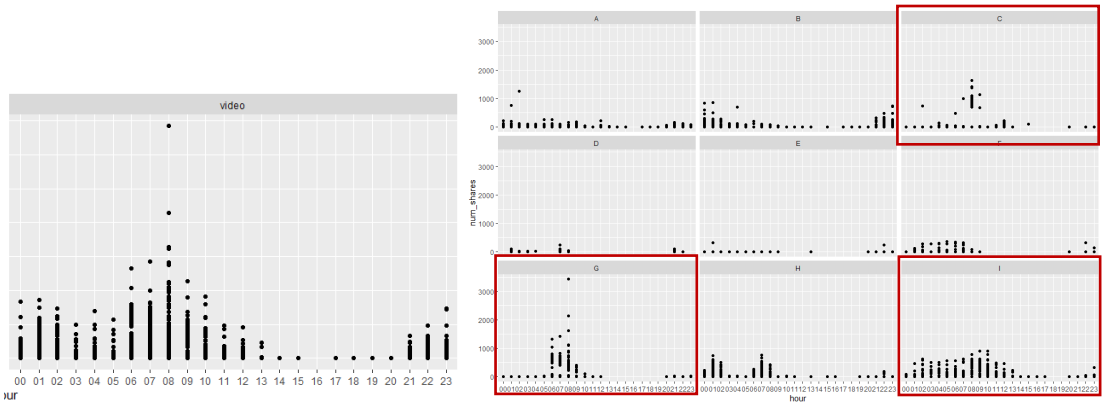
fashion companies in Thailand, they need to promote themselves more to reach potential customers. If video post viewers are happy to share the post. It is a spontaneous behavior of direct viewers and it will have a radiation effect. Even though it is a video advertisement, the indirect viewers (who watched on the share's link) will feel much better than watch it on Facebook by random advertisement recommendation. In recent days, a short clip from Tik Tok is popular and expanding very fast. Companies should take this change and adjust the marketing and promotion logic to focus more on video content. (Tables below shows numbers of shares changed for 4 types of status within date from 2012 to 2018, and within time in 1 day. Video, the red rectangular frame contributes dramatically.)




For the companies' part, there is no data about followers or other information for further analysis. However, there are still some insights. Only company A has post before emoji-in, which means when analyzing the effect of emoji_in, it could only conduct on company A. Other companies will be not influenced. And only companies A, B, I have sent links, most of the companies have sent videos. But for some "old users" of Facebook, like companies A and B, the amounts of photos are much large. For newcomers like company E, F, they only post videos.




Post hour influenced a lot in shares. It shows that shares of videos in the morning especially 8:00-9:00, will be dramatically high. Time from 14:00 to 21:00 should be avoided. From the tables below, the shape of mapping Company G, C, I, looks similar to the left one. They seldom post during 14:00 – 21:00. Previously from Random Forest, Company G, C, I, also contributed lots on the predictable model.




In a nutshell, my project can be further improved in three major areas as listed below.
Fine-tune every part of the system: The machine learning model can be further improved by finetuning. For instance, Random Forest parameters are chosen using the Randomized-search and only check OOB error with m-try and set the n-tree = 100 due to the Gird-search being too time-consuming. Perhaps I can spend more time finding the best hyperparameters. Same as SVM.

More Facebook metrics types: The result of Random Forest shows that the company will be a significant variable since it has a different size of followers. Besides, metrics about geolocation should be considered. If the geolocation of every user engagement could be determined, it will be easy for the regional analysis. Although the dataset is focused on Thai companies, Facebook is a global platform. People wherever the world could take the engagement of the post anytime. It will affect the data of the hour. For instance, there is a jet lag between the eastern and western hemispheres.

Define the transformation rate of social media engagement to actual purchasing: The metrics like the category of the post including promotion, activities, sale, product introduction should be added. In addition, there will be some metrics represent that if customers click on the link in the post and make a purchase, it could be easily recognized this deal coming from Facebook. There will be a big difference if the company purchases a Facebook ad. From the dataset, it is not shown whether these companies purchase Facebook ad service.

**Appendix**

**Linear Regression Model**

```
> summary(model.regression)

Call:
lm(formula = num_shares ~ company_B + company_C + company_D +
    company_E + company_F + company_G + company_H + company_I +
    status_type_status + status_type_video + hour_00 + hour_02 +
    hour_03 + hour_04 + hour_05 + hour_08 + hour_09 + hour_11 +
    hour_12 + hour_13 + hour_14 + hour_20 + hour_21 + hour_22 +
    hour_23, data = trainingSet_IV_DV, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-367.79  -33.93    0.12   26.72 3052.21

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         -10.186      4.222  -2.412 0.015890 *
company_B            26.723      4.949   5.399 7.00e-08 ***
company_C           208.703     12.743  16.378  < 2e-16 ***
company_D           -54.285     18.631  -2.914 0.003587 **
company_E          -111.923     11.415  -9.805  < 2e-16 ***
company_F           -92.575      9.967  -9.288  < 2e-16 ***
company_G           227.508      9.836  23.129  < 2e-16 ***
company_H            27.323      4.920   5.553 2.95e-08 ***
company_I            58.988      5.864  10.060  < 2e-16 ***
status_type_status  -32.745      8.049  -4.068 4.82e-05 ***
status_type_video   131.258      4.071  32.242  < 2e-16 ***
hour_00             -17.218      7.949  -2.166 0.030354 *
hour_02             -15.270      6.454  -2.366 0.018016 *
hour_03             -28.688      7.766  -3.694 0.000223 ***
hour_04             -15.537      8.096  -1.919 0.055043 .
hour_05             -30.898      8.257  -3.742 0.000185 ***
hour_08              23.210      6.096   3.807 0.000142 ***
hour_09             -14.113      7.600  -1.857 0.063376 .
hour_11             -24.293     12.156  -1.998 0.045720 *
hour_12             -65.667     15.318  -4.287 1.85e-05 ***
hour_13             -51.659     20.462  -2.525 0.011614 *
hour_14             -53.910     31.014  -1.738 0.082228 .
hour_20             -28.256     19.896  -1.420 0.155614
hour_21             -22.651     12.125  -1.868 0.061800 .
hour_22             -22.258      7.395  -3.010 0.002628 **
hour_23             -17.262      7.446  -2.318 0.020467 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 119 on 4945 degrees of freedom
Multiple R-squared:  0.338,     Adjusted R-squared:  0.3346
F-statistic:   101 on 25 and 4945 DF,  p-value: < 2.2e-16
```

```
> summary(model.regression2)

Call:
lm(formula = num_shares ~ emoji_in + company_B + company_C +
    company_D + company_E + company_F + company_G + company_H +
    company_I + status_type_status + status_type_video + hour_00 +
    hour_02 + hour_03 + hour_04 + hour_05 + hour_08 + hour_09 +
    hour_11 + hour_12 + hour_13 + hour_21 + hour_22 + hour_23 +
    emoji_in:status_type_video + emoji_in:status_type_status,
    data = trainingSet_IV_DV, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-384.14  -22.05    0.93   15.34 3035.86


Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                   10.635      4.731   2.248 0.024611 *
emoji_in                     -33.956      6.116  -5.552 2.98e-08 ***
company_B                     31.180      6.053   5.152 2.68e-07 ***
company_C                    206.734     12.889  16.040  < 2e-16 ***
company_D                    -70.175     18.424  -3.809 0.000141 ***
company_E                   -138.121     11.702 -11.803  < 2e-16 ***
company_F                   -119.391     10.410 -11.469  < 2e-16 ***
company_G                    217.968     10.248  21.269  < 2e-16 ***
company_H                     23.543      5.989   3.931 8.57e-05 ***
company_I                     57.371      6.785   8.456  < 2e-16 ***
status_type_status             4.042     15.630   0.259 0.795958
status_type_video              5.479      8.193   0.669 0.503676
hour_00                      -13.075      7.683  -1.702 0.088834 .
hour_02                      -11.567      6.233  -1.856 0.063567 .
hour_03                      -22.634      7.505  -3.016 0.002576 **
hour_04                      -13.812      7.819  -1.766 0.077396 .
hour_05                      -20.415      7.990  -2.555 0.010645 *
hour_08                       24.970      5.880   4.246 2.21e-05 ***
hour_09                      -11.595      7.337  -1.580 0.114096
hour_11                      -17.300     11.773  -1.469 0.141789
hour_12                      -56.532     14.834  -3.811 0.000140 ***
hour_13                      -35.378     19.845  -1.783 0.074687 .
hour_21                      -23.895     11.738  -2.036 0.041828 *
hour_22                      -22.420      7.148  -3.137 0.001719 **
hour_23                      -11.300      7.194  -1.571 0.116318
emoji_in:status_type_video   163.046      9.334  17.468  < 2e-16 ***
emoji_in:status_type_status  -43.611     18.033  -2.418 0.015625 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 115.3 on 4944 degrees of freedom
Multiple R-squared:  0.3785,    Adjusted R-squared:  0.3752
F-statistic: 115.8 on 26 and 4944 DF,  p-value: < 2.2e-16
```

**SVM Model**

```
> summary(model_svm)

Call:
svm.default(x = trainingSet_IV, y = trainingSet$num_shares,
    kernel = "linear")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.02857143
    epsilon:  0.1


Number of Support Vectors:  2155

> coef(model_svm)
       (Intercept)           emoji_in          company_B
     -1.889654e-01       2.948328e-03       7.057489e-02
         company_C          company_D          company_E
      2.719362e-02       7.643911e-03      -2.134450e-02
         company_F          company_G          company_H
     -2.503936e-02       4.168904e-02       7.705919e-02
         company_I   status_type_link status_type_status
      5.900759e-02       6.650254e-05      -1.441564e-04
 status_type_video            hour_00            hour_01
      1.520612e-01      -8.885234e-05       5.317690e-05
           hour_02            hour_03            hour_04
     -3.372120e-05      -1.554647e-03      -5.614614e-05
           hour_05            hour_06            hour_08
     -1.461904e-03       4.531938e-05       2.232926e-05
           hour_09            hour_10            hour_11
     -8.695235e-05      -1.086044e-03      -8.529256e-04
           hour_12            hour_13            hour_14
     -7.206512e-04      -5.658871e-04      -4.062581e-03
           hour_15            hour_16            hour_17
     -7.728540e-14       9.471590e-16       3.682089e-13
           hour_18            hour_19            hour_20
     -1.522256e-04      -3.844267e-04      -6.161738e-15
           hour_21            hour_22            hour_23
     -5.933748e-05      -6.437924e-05      -3.802771e-05
```

```
> summary(model_svm2)

Call:
svm.default(x = trainingSet_IV, y = trainingSet$num_shares,
 kernel = "polynomial",
    degree = "2")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  polynomial
       cost:  1
     degree:  2
      gamma:  0.02857143
     coef.0:  0
    epsilon:  0.1


Number of Support Vectors:  1637
```

**Random Forest**

```
> model_RF

Call:
 randomForest(x = trainingSet_IV, y = trainingSet$num_shares,
 ntree = 100, mtry = 15, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 15

          Mean of squared residuals: 9941.698
                    % Var explained: 53.25
```

```
> round(importance(model_RF), 2)
                     %IncMSE IncNodePurity
emoji_in               9.42    1783475.80
company_B             17.33    1182549.49
company_C             27.52    9627141.98
company_D              3.71      29424.09
company_E              6.92     724144.39
company_F              3.10     655443.59
company_G             38.75   12633788.48
company_H             29.63    2231412.65
company_I             37.15    3012434.11
status_type_link       3.42       7461.85
status_type_status     3.47     486311.75
status_type_video     49.09   18036101.50
hour_00                7.09     195433.10
hour_01                2.94     689744.82
hour_02                3.15     359294.96
hour_03                8.23     295532.82
hour_04                2.26     266646.87
hour_05                1.39     144957.16
hour_06               -0.54     515240.92
hour_08                7.89    3427649.96
hour_09                8.66    1193437.65
hour_10                3.74     245687.44
hour_11                0.85     243157.44
hour_12                6.85    1135093.63
hour_13                2.50      63927.18
hour_14                1.40      27896.64
hour_15                0.83       6222.49
hour_16                2.04         17.97
hour_17                0.00         16.31
hour_18                3.51         16.18
hour_19                4.12       1205.83
hour_20                4.56       8253.11
hour_21                9.51     184421.95
hour_22               13.73     425284.60
hour_23                4.45      97929.89
```

```
> model_RF4

Call:
 randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1])
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 15

          Mean of squared residuals: 9940.051
                    % Var explained: 53.26
```

```
> round(importance(model_RF4), 2)
                    IncNodePurity
emoji_in              1855487.41
company_B             1147026.17
company_C             8957791.29
company_D               49666.09
company_E              804950.55
company_F              783532.42
company_G            12039312.83
company_H             2297937.88
company_I             2900140.18
status_type_link         7314.30
status_type_status     499881.36
status_type_video    19036904.59
hour_00                210102.82
hour_01                669889.74
hour_02                328185.51
hour_03                307820.11
hour_04                307643.15
hour_05                143152.19
hour_06                419529.77
hour_08               3041606.28
hour_09               1309104.73
hour_10                282739.59
hour_11                197095.86
hour_12               1454986.09
hour_13                 63672.96
hour_14                 33924.55
hour_15                  5639.95
hour_16                    18.52
hour_17                    28.71
hour_18                    24.66
hour_19                  1315.08
hour_20                  7500.83
hour_21                184411.00
hour_22                472953.39
hour_23                105303.78
```

**Prediction Error**

RF4-Light blue

SVM2-Purple

Linear Regression2-Pink