R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

PCA and LDA reduction

Vedant Prakash

Load the Data

Hide

```
df <- job_profitability
df <- df[,c(3, 4, 5, 10)]
str(df)
```

```
'data.frame':   14479 obs. of  4 variables:
 $ Jobs_Gross_Margin: num  -4.01 254.13 151.83 -32.15 222.7 ...
 $ Labor_Pay        : num  0 91 0 0 0 ...
 $ Labor_Burden     : num  22.2 14.9 133.2 81.2 66.3 ...
 $ Jobs_Total       : num  79.5 360 289 49 289 ...
```

Divide into train and test

Hide

```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df) * 0.8, replace = FALSE)

train <- df[i, ] # training data
test <- df[-i, ] # testing data
```

Run PCA

Hide

```
library(caret)
pca_out <- preProcess(train[, -1], method=c("center", "scale", "pca"))
pca_out
```

```
Created from 11583 samples and 3 variables

Pre-processing:
  - centered (3)
  - ignored (0)
  - principal component signal extraction (3)
  - scaled (3)

PCA needed 3 components to capture 95 percent of the variance
```

Hide

```
train_pc <- predict(pca_out, train[, -1])
test_pc <- predict(pca_out, test[,-1])
train_pc
```

	PC1 <dbl>	PC2 <dbl>	PC3 <dbl>
7452	0.2274677569	0.0478347080	-0.2702621219
8016	-0.6597695635	-0.1871328438	0.2278626458
7162	-0.3295522603	0.2659012593	0.1762793848
8086	0.2086231556	-0.8051990915	0.0553262420
7269	0.9392737785	-0.3507928155	0.4278437286
9196	0.2577043871	0.7608296199	0.0930945944
623	-0.4461315589	0.0761324687	0.1207905256
10885	0.3094015708	-0.0208571302	-0.1844834396
934	-0.1698675808	-0.5227438034	-0.0027214215
12688	-0.7727363954	-0.0366094770	-0.1014314329

1-10 of 11,583 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
NA
```

Using PCA principal components to predict

Here we will use the two prinicipal components instead of the three predictors to predict

Hide

```
train_df <- data.frame(PC1=train_pc$PC1, PC2=train_pc$PC2, Gross_Margin=train$Jobs_Gross_Margin)
test_df <- data.frame(PC1=test_pc$PC1, PC2=test_pc$PC2, Gross_Margin=test$Jobs_Gross_Margin)
lm1 <- lm(train_df$Gross_Margin ~ ., data = train_df) # builds linear regression model
summary(lm1) # shows the linear regression model summary
```

```

Call:
lm(formula = train_df$Gross_Margin ~ ., data = train_df)

Residuals:
    Min      1Q  Median      3Q     Max 
-7778.4   -57.3    -9.2    64.5 11665.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  296.418      2.452   120.9   <2e-16 ***
PC1          237.854      1.648   144.3   <2e-16 ***
PC2          817.100      3.239   252.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 263.9 on 11580 degrees of freedom
Multiple R-squared:  0.8795,    Adjusted R-squared:  0.8794 
F-statistic: 4.224e+04 on 2 and 11580 DF,  p-value: < 2.2e-16
```

Accuracy on Test Data

Hide

```
# Evaluate accuracy on test
pred <- predict(lm1, newdata = test_df)
cor_lm <- cor(pred, test_df$Gross_Margin)
mse_lm <- mean((pred - test_df$Gross_Margin)^2)
cor_lm
```

```
[1] 0.954364
```

Hide

```
mse_lm
```

```
[1] 40315.72
```

Hide

```
print(paste('rmse:', sqrt(mse_lm)))
```

```
[1] "rmse: 200.787747189924"
```

The regression with all 3 features received a correlation of 0.957191866567773,mse of 38144.5006673951, and rmse of 195.306171606007 compared to the regression with the reduced data which received a correlation of 0.954, mse of 40315.72, and rsme of 200.78775.

The regression with all 3 features had a higher correlation coefficient (0.957) compared to the regression with reduced data (0.954). This means that the regression with all 3 features had a stronger linear relationship between the independent and dependent variables.

Moreover, the regression with all 3 features also had a lower mean squared error (MSE) of 38,144.50, compared to the MSE of 40,315.72 for the regression with reduced data. This indicates that the predictions of the regression with all 3 features were, on average, closer to the actual values.

Finally, the root mean squared error (RMSE) for the regression with all 3 features was 195.31, which was lower than the RMSE of 200.79 for the regression with reduced data. RMSE is a more interpretable metric since it's on the same scale as the original data, and a lower RMSE indicates that the model's predictions were more accurate.

KNN regression compare PCA

Hide

```
#Scale the data
train_scaled <- train_df[, 1:2]
means <- sapply(train_scaled, mean)
stdvs <- sapply(train_scaled, sd)
train_scaled <- scale(train_scaled, center = means, scale = stdvs)
test_scaled <- scale(test_df[, 1:2], center = means, scale = stdvs)
# Fit the model (using k value found in part 1)
fit_17 <- knnreg(train_scaled, train_df$Gross_Margin, k = 17)
# Evaluate
predictions_17 <- predict(fit_17, test_scaled)
cor_knn17 <- cor(predictions_17, test_df$Gross_Margin)
mse_knn17 <- mean((predictions_17 - test_df$Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_knn17))
```

```
[1] "cor: 0.9539637919202"
```

Hide

```
print(paste('mse:', mse_knn17))
```

```
[1] "mse: 40588.338950722"
```

Hide

```
print(paste('rmse:', sqrt(mse_knn17)))
```

```
[1] "rmse: 201.465478310111"
```

The data reduced by PCA lost accuracy. It has a correlation of 0.95396, a mse of 40588.33895, and an rsme of 201.465478. The data with all three features had a correlation of 0.95719, a mse of 38144.500667, and a rsme of 195.30617.

Decision Tree compare PCA

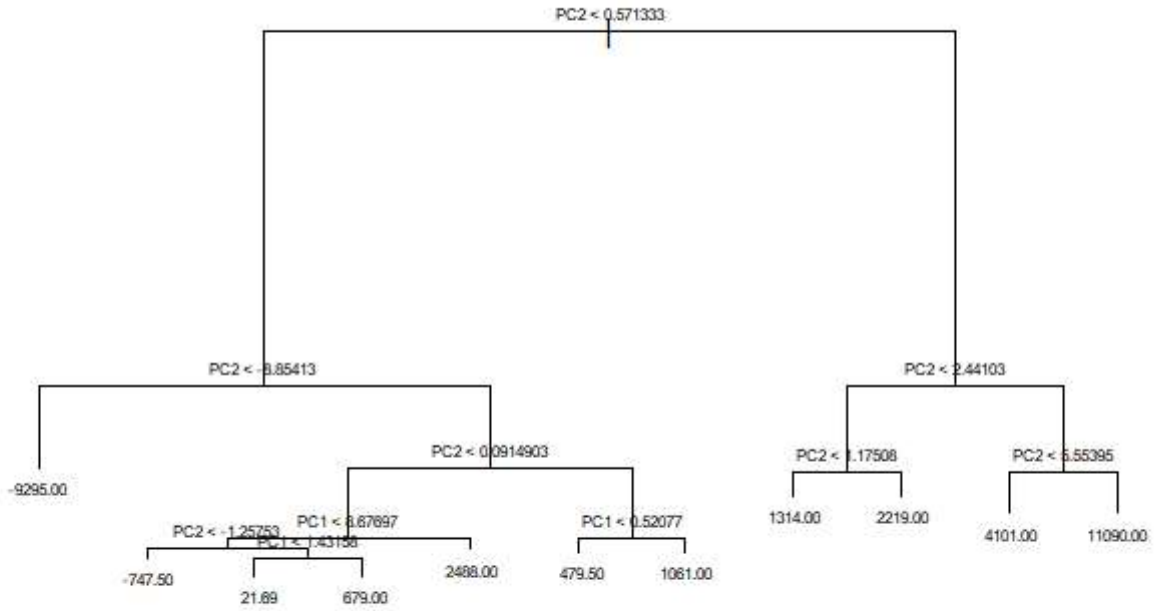
Hide

```
#Initial tree

library(tree)
library(MASS)
# Build tree
tree1 <- tree(Gross_Margin~., data=train_df)
summary(tree1)
```

```

Regression tree:
tree(formula = Gross_Margin ~ ., data = train_df)
Number of terminal nodes:  11
Residual mean deviance:  113500 = 1.313e+09 / 11570
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-6694.00  -129.10   -43.17     0.00   123.50  8354.00
```

Hide

```
plot(tree1)
text(tree1, cex = 0.5, pretty = 0)
```



Hide

```
# Evaluate tree
pred <- predict(tree1, newdata=test_df)
cor_tree <- cor(pred, test_df$Gross_Margin)
mse_tree <- mean((pred-test_df$Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_tree))
```

```
[1] "cor: 0.880403870206271"
```

Hide

```
print(paste('mse:', mse_tree))
```

```
[1] "mse: 103925.141733997"
```

Hide

```
print(paste("rmse :", sqrt(mse_tree)))
```

```
[1] "rmse : 322.374226224736"
```

The data reduced by PCA lost accuracy. It has a correlation of 0.8804, a mse of 103925.142, and an rsme of 322.374. The data with all three features had a correlation of 0.914753236805949, a mse of 72983.9813219516, and a rsme of 270.155476202041.

Pruning

Hide

```
tree_pruned <- prune.tree(tree1, best = 5)
# Evaluate
pred_pruned <- predict(tree_pruned, newdata=test_df)
cor_pruned <- cor(pred_pruned, test_df$Gross_Margin)
mse_pruned <- mean((pred_pruned-test_df$Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_pruned))
```

```
[1] "cor: 0.813053111529781"
```

Hide

```
print(paste('mse:', mse_pruned))
```

```
[1] "mse: 153362.028372789"
```

Hide

```
print(paste('rmse:', sqrt(mse_pruned)))
```

```
[1] "rmse: 391.614642694562"
```

The results of the PCA-reduced dataset after pruning are shown. The original data received a cor of 0.859246727417677, a mse of 116650.137250333, and a rsme of 341.54082808697 after pruning.

Random Forest

Hide

```
library(randomForest)
```

```
randomForest 4.7-1.1
Type rfNews() to see new features/changes/bug fixes.

Attaching package: ‘randomForest’

The following object is masked from ‘package:ggplot2’:

    margin
```

Hide

```
set.seed(1234)
rf <- randomForest(Gross_Margin~., data = train_df, importance = TRUE)
rf
```

```
Call:
 randomForest(formula = Gross_Margin ~ ., data = train_df, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 94201.87
                    % Var explained: 83.69
```

Hide

```
# Evaluate
pred_rf <- predict(rf, newdata = test_df)
cor_rf <- cor(pred_rf, test_df$Gross_Margin)
mse_rf <- mean((pred_rf-test_df$Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_rf))
```

```
[1] "cor: 0.951514023736301"
```

Hide

```
print(paste('mse:', mse_rf))
```

```
[1] "mse: 42269.6035215972"
```

Hide

```
print(paste('rmse:', sqrt(mse_rf)))
```

```
[1] "rmse: 205.595728364179"
```

The results for the PCA-reduced data for random forest are shown. The orginal data received a cor of 0.956297620429714, a mse of 38520.938978973, and a rsme of 196.267518909709.

Bagging

Hide

```
bag <- randomForest(Gross_Margin~., data=train_df, mtry = 2)
bag
```

```
Call:
 randomForest(formula = Gross_Margin ~ ., data = train_df, mtry = 2)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 81470.97
                    % Var explained: 85.89
```

Hide

```
# Evaluate
pred_bag <- predict(bag, newdata=test_df)
cor_bag <- cor(pred_bag, test_df$Gross_Margin)
mse_bag <- mean((pred_bag-test_df$Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_bag))
```

```
[1] "cor: 0.949809766257569"
```

Hide

```
print(paste('mse:', mse_bag))
```

```
[1] "mse: 43959.4678500128"
```

Hide

```
print(paste('rmse:', sqrt(mse_bag)))
```

```
[1] "rmse: 209.665132652076"
```

The accuracy for the PCA-reduced data is shown. The original data had an cor of 0.957401702050157, a mse of 37216.2941883724, and a rsme of 192.915251310964 for bagging.

LDA reduction

Now we will move on to LDA reduction and compare with the original data again.

Hide

```
library(MASS)
lda1 <- lda(Jobs_Gross_Margin~., data=train)
lda1$means
```

	Labor_Pay	Labor_Burden	Jobs_Total
-11522.96	1900.41000	4866.56000	0.00000
-10971.98	439.06000	5940.65000	0.00000
-10103.68	1808.79000	4457.98000	0.00000
-9612.99	1858.99000	4243.39000	0.00000
-8424.62	2283.09000	2507.28000	0.00000
-7334.45	2233.55000	1553.38000	0.00000
-7092.73	2696.77000	2578.71000	0.00000
-5076.7	955.52000	305.55000	0.00000
-3700.27	63.05000	25.75000	120.00000
-2601.53	359.01000	379.52000	0.00000
-2406.44	392.99000	414.71000	0.00000
-2362.15	174.07000	155.60000	0.00000
-2348.96	651.74000	739.92000	0.00000
-2232.65	334.41000	390.93000	49.00000
-2007.32	665.20000	884.59000	0.00000
-1900.89	205.14000	118.62000	1197.70000
-1811.79	873.94000	937.85000	0.00000
-1766.45	334.76000	188.62000	0.00000
-1757.81	1061.35000	736.37000	655.32000
-1730.33	640.77000	983.30000	0.00000
-1703.81	980.84000	225.10000	0.00000
-1689.07	1130.97000	313.15000	0.00000
-1597.42	906.51000	868.83000	718.25000
-1499.56	262.68000	301.52000	0.00000
-1480.32	241.67000	281.55000	0.00000
-1426.72	488.64000	592.58000	0.00000
-1325.17	801.45000	845.59000	2218.56000
-1312.64	542.55000	526.37000	0.00000
-1303.48	611.59000	462.33000	0.00000
-1286.51	346.30000	499.36000	0.00000
-1280.92	166.72000	199.75000	0.00000
-1261.12	508.37000	632.61000	0.00000
-1246.25	781.17000	314.98000	0.00000
-1186.32	559.05000	564.04000	0.00000
-1171.52	635.73000	102.81000	0.00000
-1147.94	234.91000	258.72000	0.00000
-1111.98	432.02000	63.73000	0.00000
-1097.83	356.51000	404.25000	838.00000
-1095.6	281.40000	20.07000	0.00000
-1087.96	382.36000	512.68000	0.00000
-1082.25	523.78000	410.70000	0.00000
-1074.5	250.78000	116.71000	0.00000
-1065.67	0.00000	170.38000	249.00000
-1060.99	290.40000	134.75000	0.00000
-1056.3	166.37000	84.44000	0.00000
-1051.21	295.55000	406.74000	0.00000
-1049.18	2531.62000	2827.82000	7152.00000
-1048.31	292.11000	118.99000	4345.64000
-1048	265.55000	316.03000	0.00000
-1035.27	593.57000	185.53000	0.00000
-1028.02	436.81000	531.70000	0.00000

```
-999.22    823.38000    868.40000  1564.15000
-976.74    563.50000     18.23000     0.00000
-974.35    341.51000    482.24000     0.00000
-947.41    720.41000    757.13000  1350.65000
-945.61    240.27000     93.02000     0.00000
-938.14    243.23000    185.42000     0.00000
-926.64   1074.92000      1.72000   150.00000
-924.29    307.36000    397.59000     0.00000
-908.6     367.07000    477.49000     0.00000
-902.9     349.17000    499.36000     0.00000
-901.03    174.43000     93.57000     0.00000
-900.16    428.25000    439.44000     0.00000
-862.6     268.11000    354.79000   139.00000
-854.67    327.89000    173.53000     0.00000
-849.11    403.43000    423.15000     0.00000
-848.51    857.73000   1054.72000  1853.00000
-834.52    395.67000    573.55000  1501.93000
-828.05    319.39000    337.66000     0.00000
-820.95    384.86000    178.02000     0.00000
-816.4     446.25000    329.10000     0.00000
-807.83    375.10000    432.73000     0.00000
-804.43    377.58000    356.18000     0.00000
-802.02    320.11000    300.57000     0.00000
-796.86    299.70000    405.44000     0.00000
-789.29    241.93000    288.20000   169.00000
-782.02    197.68000    264.42000     0.00000
-780.49    206.18000    246.35000     0.00000
-778.47    203.14000    274.89000     0.00000
-778.42    355.15000    423.27000     0.00000
-773       212.57000    280.59000     0.00000
-769.66    366.02000    403.64000     0.00000
-765.73     93.69000     77.04000     0.00000
-761.38    390.88000    370.50000     0.00000
-745.98    220.86000    293.91000     0.00000
-735.65    342.47000    434.68000     0.00000
-733.86    337.75000    314.84000     0.00000
-732.43    246.20000    361.43000   199.00000
-725.99    148.69000    209.04000     0.00000
-723.26    220.19000    231.13000     0.00000
-722.66    310.86000    259.59000     0.00000
-722.34    286.57000    128.44000     0.00000
-719.08    292.04000    358.59000     0.00000
-702.71    354.37000    157.89000     0.00000
-694.86    583.30000     19.61000     0.00000
-694.1     362.86000    269.18000     0.00000
-686.09      0.00000    164.40000     0.00000
-680.35    264.51000     98.48000     0.00000
-678.44    228.52000    107.48000     0.00000
-674.24    200.34000    272.03000     0.00000
-669.56    333.13000    255.86000     0.00000
-668.32    309.62000    282.50000     0.00000
-659.13    299.46000    332.91000   463.25000
```

```
-657.47    249.27000    225.43000     0.00000
-656.35    278.49000    115.57000     0.00000
-655.47    328.74000    296.43000     0.00000
-642.03    107.88000    117.94000     0.00000
-638.97    197.01000    280.73000     0.00000
-638.63    477.70000    151.71000     0.00000
-638.12    278.51000    127.93000     0.00000
-637.81    521.35000    116.46000     0.00000
-636.7     195.82000    227.33000     0.00000
-627.14    293.90000    306.28000     0.00000
-622.56    429.86000    177.50000     0.00000
-618.92    182.57000    200.70000   249.00000
-610.87    252.04000     69.44000     0.00000
-607.65    182.56000    397.39000     0.00000
-607.02    271.31000    335.71000     0.00000
-606.56    121.03000     53.47000     0.00000
-604.96    290.12000    314.84000     0.00000
-600.87    218.55000    251.11000   642.00000
-596.71    161.11000    206.40000     0.00000
-593.27    241.30000    269.18000     0.00000
-592.22    280.23000    311.98000     0.00000
-592.17    124.69000    140.77000     0.00000
-587.31     93.55000    135.07000     0.00000
-585.26    200.44000    264.42000     0.00000
-583.93    244.47000    307.23000     0.00000
-578.65    410.03000    122.62000     0.00000
-578.09    178.09000    147.43000     0.00000
-577.87    193.34000    229.23000     0.00000
-574.96    254.52000    250.51000     0.00000
-574.77    239.01000    335.76000     0.00000
-573.74    188.90000     75.21000     0.00000
-571.96    135.96000    169.31000     0.00000
-569.93    205.86000    261.57000     0.00000
-569.35    283.00000    331.01000    89.00000
-569.17    691.09000    874.67000  1231.65000
-568.73     80.60000     32.88000   240.00000
-565.37    116.82000    136.89000     0.00000
-557.71    264.09000    343.37000     0.00000
-556.87    135.97000    131.26000     0.00000
-549.73    164.69000    129.36000     0.00000
-548.86    102.13000    135.07000     0.00000
-543.42    117.46000    118.30000   199.00000
-540.33    101.25000     20.16000     0.00000
-534.23    924.63000    927.39000  1955.30000
-533.89    186.72000    147.43000     0.00000
-533.08    163.22000    182.00000     0.00000
-529.5     314.58000    214.92000   249.00000
-526.84    561.00000     74.54000  1101.92000
-526.39      0.00000    180.22000   199.00000
-526.25    149.56000    331.93000     0.00000
-522.38    334.99000    145.94000   445.00000
-521.53    289.45000    232.08000     0.00000
```

```
-520.45    134.33000     64.03000     0.00000
-519.52     89.67000     55.17000     0.00000
-519.49     44.68000     38.05000    49.00000
-517.19    212.25000    258.00000     0.00000
-516.87    328.11000    186.43000     0.00000
-516.85    119.78000    154.09000     0.00000
-515.79    205.95000     79.66000     0.00000
-515.78    297.01000    218.77000   139.00000
-514.71    231.51000    275.00000     0.00000
-514.46    230.44000    286.30000     0.00000
-514.09     63.84000     55.15000     0.00000
-514.01    225.65000    272.98000     0.00000
-513.51    145.94000     48.57000     0.00000
-511.57     86.18000     98.92000   199.00000
-501.2     344.69000    156.51000     0.00000
-500.86    134.00000     31.55000     0.00000
-500.45    330.02000    170.43000     0.00000
-499.8     208.74000    291.06000     0.00000
-498.47    152.85000    191.18000     0.00000
-498.27    362.86000    286.30000   254.85000
-497.14    184.48000     89.36000     0.00000
-497.13    193.34000    236.36000     0.00000
-496.1     218.63000    170.26000     0.00000
-494.95    238.13000    256.82000     0.00000
-492.16    109.35000     49.20000     0.00000
-490.17    245.44000    291.06000     0.00000
-486.47    102.14000    120.80000     0.00000
-485.23    204.28000    226.70000     0.00000
-484.31     97.67000    121.75000     0.00000
-482.13    216.69000    230.18000     0.00000
-480.52    176.82000    116.99000     0.00000
-479.22    371.26000     61.06000     0.00000
-478.61    326.95000    200.66000   248.00000
-475.55     77.95000     63.62000     0.00000
-475.05    347.59000    127.46000     0.00000
-474.38    222.32000    252.06000     0.00000
-468.25    216.08000    147.43000   139.00000
-463.36    230.32000    233.04000     0.00000
-461.86    129.59000    234.94000   -50.00000
-459.94      0.00000    337.80000   372.75000
-454.92    124.55000     59.16000     0.00000
-452.15    195.33000    256.82000     0.00000
-450.75    209.15000    241.60000     0.00000
-450.12    180.92000    205.45000     0.00000
-446.33    138.00000    149.33000     0.00000
-446.24    179.37000    172.16000     0.00000
-442.05    139.97000    195.91000     0.00000
-437.73    206.29000    231.44000     0.00000
-437.33    303.13000    134.20000     0.00000
-437.32    233.80000    156.52000     0.00000
-435.24    150.21000    187.38000     0.00000
-435.12    201.76000    233.36000   388.00000
```

```
-435.1     186.68000    248.42000     0.00000
-434.29    111.18000     56.42000     0.00000
-433.31    248.30000    154.67000     0.00000
-432.71    324.35000    108.36000     0.00000
-432.03    198.24000    210.21000     0.00000
-431.57    170.92000    196.89000     0.00000
-430.63    283.40000     22.44000     0.00000
-426.58    104.05000    116.61000   279.25000
-426.31    183.76000    242.55000     0.00000
-424.86    150.97000    273.89000     0.00000
-423.71    119.34000     44.80000     0.00000
-422.52    330.00000     92.52000     0.00000
-422.3     187.91000    234.39000     0.00000
-420.69    203.15000    204.50000     0.00000
-420.64    199.64000    221.00000     0.00000
-414.4     132.77000    101.77000     0.00000
-410.69    218.90000    130.43000     0.00000
-408.28    272.49000    135.79000     0.00000
-404.79    210.85000    191.18000     0.00000
-403.88    341.61000     27.42000     0.00000
-398.01    255.33000    142.68000     0.00000
-397.77    195.89000     77.27000   300.00000
-397.76    170.43000    227.33000     0.00000
-396.71    261.64000    135.07000     0.00000
-396.19     35.37000      6.06000     0.00000
-395.76      0.00000    127.99000     0.00000
-393.48    147.30000     93.00000     0.00000
-393.25    255.43000    110.62000     0.00000
-392.74    438.26000    551.07000   819.50000
-392.7     176.82000    188.33000     0.00000
-390.47     98.51000     43.40000     0.00000
-389.07    216.91000    172.16000     0.00000
-382.75    208.74000     88.46000     0.00000
-379.91    179.75000     81.86000     0.00000
-379.78    173.94000    231.44000   138.71000
-379.48    151.99000     40.76000   519.00000
-378.57    200.35000    219.72000     0.00000
-377.5     183.76000    116.99000   463.25000
-377.31    230.44000    260.62000   503.00000
-374.36    347.92000    278.69000   609.00000
-373.92    160.86000    213.06000     0.00000
-373.8     188.32000    185.48000     0.00000
-372.92    147.81000    200.70000     0.00000
-372.46    176.52000    195.94000     0.00000
-372.01    153.89000    218.12000     0.00000
-371.27    301.34000     69.93000     0.00000
-370.78    165.33000    205.45000     0.00000
-370.12    160.86000    209.26000   139.00000
-370.02    128.63000    150.28000     0.00000
-369.7       0.00000    130.00000     0.00000
-369.2      79.15000    290.05000     0.00000
-367.92    163.42000    204.50000     0.00000
```

```
-367.08    146.18000    168.16000     0.00000
-366.46    113.98000    128.41000     0.00000
-365.95    177.11000    183.58000     0.00000
-365.7     141.25000    161.70000     0.00000
-365.54    311.41000     18.78000     0.00000
-363.62    210.88000    152.74000     0.00000
-363.25    173.02000    190.23000     0.00000
-362.28    187.43000     84.48000     0.00000
-361.64     42.67000      3.21000   383.40000
-357.79    550.33000    485.00000   964.99000
-356.57    454.56000    506.02000  1378.80000
-356.25    114.67000      1.83000    49.00000
-356.1       0.00000    116.21000     0.00000
-354.37    242.84000    111.53000     0.00000
-353.37    254.21000    101.67000     0.00000
-352.18    196.18000    156.00000   199.00000
-351.41    185.65000     62.76000     0.00000
-351.36    174.44000    176.92000     0.00000
-351.27     98.83000    113.10000     0.00000
-347.98     98.07000    107.48000    49.00000
-347.92     84.26000     76.05000     0.00000
-347.39    512.42000    537.41000     0.00000
-346.8     163.22000    183.58000     0.00000
-345.07    221.64000    172.43000   248.00000
-345       174.46000    216.87000   139.00000
-344.08    219.21000     91.57000     0.00000
-343.86    167.89000    175.97000   199.00000
-342.38    214.35000    128.03000     0.00000
-339.39    256.94000     82.45000     0.00000
-338.7     192.75000    185.48000   199.00000
-337.83    171.08000    143.63000     0.00000
-337.17    151.69000    185.48000     0.00000
-334.71    150.65000    174.06000   398.00000
-334.22    148.74000    185.48000     0.00000
-334.15    128.19000    153.63000     0.00000
-333.62    157.67000    168.36000   249.00000
-333.07    182.57000    194.99000     0.00000
-332.42    215.13000    112.76000     0.00000
-329.42    128.73000    137.92000     0.00000
-329.27    153.19000    176.08000     0.00000
-329.03    230.76000     88.13000     0.00000
-328.78    186.67000     29.88000     0.00000
-327.27    181.03000     71.93000     0.00000
-325.26    209.26000    116.00000     0.00000
-324.91    164.69000     69.00000     0.00000
-322.97    222.15000     69.18000     0.00000
-321.9     234.27000     87.63000     0.00000
-321.66    216.98500    117.09000     0.00000
-319.7     231.72000     87.98000     0.00000
-319.21    165.96000    120.17000   487.00000
-318.99    145.01000    162.65000     0.00000
-318.04    215.23000    102.81000     0.00000
```

```
-317.87    162.83000    155.04000     0.00000
-317.04    103.41000     22.83000     0.00000
-316.82    176.81000    136.02000   199.00000
-316.15    215.12000    249.16000   628.14000
-315.92    257.38000     83.37000     0.00000
-314.88    156.71000    204.50000     0.00000
-314.61    247.68000     66.93000   199.00000
-313.86    196.86000    117.00000     0.00000
-312.21    241.14000     71.07000   448.00000
-310.39    249.86000     40.30000     0.00000
-310.25    309.64000    136.86000   377.00000
-309.99    139.73000    170.26000     0.00000
-309.96    224.07000     85.89000     0.00000
-309.9     152.01000    157.89000     0.00000
-309.34    123.29000    135.07000     0.00000
-309.33    123.20000    134.11000     0.00000
-308.71    221.71000    136.00000    49.00000
-308.61    145.01000    163.60000     0.00000
-307.94    154.80000    153.14000   139.00000
-307.23     91.28000    119.16000     0.00000
-306.64    129.58000    158.84000     0.00000
-306.33    201.33000    154.00000   248.00000
 [ reached getOption("max.print") -- omitted 10542 rows ]
```

Hide

```
lda_pred <- predict(lda1, newdata=test, type="class")
lda_pred$class
```

```
  [1] 164.51   -48.5    -26.16   -137.88  -526.84  288.17   -33.03   -68.2    -16.06   1277.78
253.19
 [12] 368.45   -16.06   2668.37  -3.18    2046.66  600.64   -98.76   555.34   -36.47   555.34
624.07
 [23] 254.13   467.2    469.89   -66.51   495.66   160.61   1047.99  499.63   234.76   -206.55
-26.38
 [34] -16.06   -16.06   415.88   -82.41   -16.06   147.83   77.77    1016.26  77.77    714.83
-16.06
 [45] -16.06   147.83   -71.42   246.6    -33.03   1210.99  -33.03   147.83   -33.03   499.63
-177.79
 [56] 317.78   535.44   228.74   282.12   104.64   -66.51   539.66   561.17   -33.03   -321.66
264.16
 [67] 3873.1   246.6    160.61   -16.06   -33.03   1412.14  147.83   -1111.98 688.4    77.77
1254.12
 [78] 228.74   2149.26  248.18   228.74   -16.06   1867.57  506.99   495.66   -396.71  -71.42
198.58
 [89] 147.83   147.83   859.98   379.62   419.39   -85.49   582.83   1079.61  254.13   264.16
264.16
[100] -16.06   -362.28  -36.47   -98.76   2148.02  -371.27  288.17   731.53   147.83   729.83
544.83
[111] 284.88   -74.43   -197.4   147.83   -16.06   1441.59  272.08   -16.06   -16.06   77.77
147.83
[122] -33.03   368.45   228.26   817.51   -98.76   -36.47   -66.51   608.11   3097.8   107.14
5128.01
[133] 817.51   237.13   147.83   -189.92  147.83   -16.06   -26.38   3157.39  -16.06   452.02
15.98
[144] -16.06   -70.82   -177.79  -98.76   452.02   288.17   323.3    -16.06   147.83   539.66
420.84
[155] 1560.08  1320.41  -71.42   582.83   499.63   1424.85  452.02   1043.66  210.1    -16.06
1071.37
[166] -226.9   714.83   -16.06   -182.45  -16.06   -33.03   210.1    -182.45  582.83   836.4
228.74
[177] 368.45   -53.63   -16.06   -396.71  -16.06   943.85   -70.1    344.43   264.16   254.13
-182.45
[188] 539.66   -71.42   -16.06   -16.06   425.16   452.02   -98.76   415.88   539.66   379.62
326.56
[199] -16.06   -98.76   -182.45  1252.09  495.66   431.79   359.57   -98.76   -182.45  714.83
4444
[210] -96.91   154.34   2002.42  425.23   716.72   147.83   1151.07  -279.35  415.88   285.73
-14.13
[221] -33.03   415.88   -3.18    1238.35  -16.06   896.33   368.45   490.66   231.38   -33.03
1075.56
[232] 368.45   -98.76   -16.06   26.85    1177.4   147.83   -114.11  152.99   288.17   -176.48
-89.68
[243] 327.96   -16.06   -48.5    403.07   555.34   121.98   -16.06   344.43   539.66   340.29
-202.42
[254] -232.22  669.75   -14.13   -70.82   -70.82   -33.03   -16.06   215.17   -144.89  -114.11
-27.56
[265] 1022.44  -33.03   1976.69  463.26   -36.47   -149.92  -174.19  499.63   -33.03   194.29
147.83
[276] 147.83   -16.06   -177.79  140.33   422.46   248.18   152.99   -265.08  -33.03   -294.98
559.18
```

```
 [287] 369.09   -33.03   -14.13   253.19   338      1917.64  431.79   1175.79  -65.22   -149.17
-583.93
 [298] 1620.06  187.61   -48.5    452.02   -33.03   499.63   1255.78  -14.13   -144.89  -16.06
-98.76
 [309] -114.11  -14.13   852.72   -333.07  -33.03   213.06   683.99   152.99   555.34   -719.08
1600.2
 [320] 665.23   -33.03   -16.06   -33.03   264.26   418.74   -16.06   147.83   561.17   147.83
152.99
 [331] 169.44   -33.03   -257.74  379.62   152.99   -16.06   -190.5   -202.42  -33.03   -14.13
-108.43
 [342] -36.47   362.3    3615.81  -174.19  -33.03   -87.53   -1097.83 -1261.12 -33.03   544.83
-202.42
 [353] -89.68   -144.89  -114.11  -14.13   676.14   115.72   -14.13   278.07   -89.68   738.3
-33.03
 [364] 147.83   -33.03   152.99   1154.95  135.42   -70.82   -36.47   1917.64  -114.11  -174.19
194.29
 [375] -36.47   242.42   -87.79   -94.33   667.11   314.98   -36.47   -16.06   169.44   -16.06
152.99
 [386] 750.56   215.17   -69.75   38.03    536.34   160.61   -36.47   -61.9    -89.68   -114.11
-14.13
 [397] -125.33  675.77   -474.38  158.57   -112.87  115.72   152.99   98       89.53    914.87
924.44
 [408] -114.11  -33.03   425.23   321.92   231.57   -571.96  -33.03   264.26   425.23   -14.13
724.51
 [419] -16.06   -70.82   -14.13   152.99   -257.74  -98.76   -257.74  665.23   -36.47   65.63
171.8
 [430] 194.29   817.51   -16.06   -33.03   340.29   665.23   1488.41  849.43   -14.13   -33.03
-14.13
 [441] 235.53   -33.03   246.6    -16.06   -33.03   -33.03   -177.79  1491.63  -112.87  -33.03
-149.17
 [452] 115.72   -14.13   675.77   679.13   544.83   152.99   -33.03   -14.13   -36.47   -14.13
-114.11
 [463] 152.99   -33.03   171.8    -14.13   340.29   -80.33   -202.42  -266.38  1617.46  231.38
-114.11
 [474] 700.56   682.39   -155.94  152.99   -254.05  -125.84  264.26   716.72   242.42   431.79
-3.18
 [485] 313.21   789.19   -114.11  194.29   -114.11  -70.82   425.23   -306.64  -337.83  675.77
-14.13
 [496] 431.79   425.23   340.29   -64.01   1721.91  -257.74  561.17   -33.03   -36.47   362.3
-110.3
 [507] -112.87  152.99   152.99   407.53   125.89   500.92   -98.76   305.61   -334.22  -53.63
152.99
 [518] 343.26   235.84   -36.47   -1048    69.69    -150.6   -33.03   245.33   67.45    452.02
-70.82
 [529] -14.13   -16.06   -127.11  -446.24  152.99   1705.36  -33.03   1.42     -114.11  -14.13
-36.47
 [540] -174.33  -279.8   135.42   74.9     675.77   -26.16   194.29   561.17   247.47   101.93
2378.22
 [551] 1086.43  -112.87  -197.4   -127.11  124.45   -392.74  -48.5    -33.03   -14.13   197.81
-61.9
 [562] -938.14  -14.13   -53.63   -14.13   152.99   797.92   362.3    -66.31   -114.11  -33.03
-61.9
```

```
 [573] -114.11  -496.1   314.76   288.17   1085.43  1158.94  -36.47   1375.98  606.01   -14.13
147.83
 [584] 379.62   2571.08  -16.06   -26.38   -16.06   -33.03   -65.77   -33.03   1930.19  -279.35
203.41
 [595] -107.8   -151.25  -36.47   561.17   579.45   242.31   -16.06   343.26   -176.48  77.93
1190.52
 [606] 743.01   817.51   1133.01  -16.06   -33.03   1240.58  -3.18    418.05   -26.38   -36.23
-14.13
 [617] -80.33   -70.82   -36.47   -14.13   -14.13   -98.76   147.83   1208.21  1016.26  -33.03
147.83
 [628] 147.83   379.62   582.83   147.83   -71.42   407.53   -337.83  456.21   -33.03   152.99
671.7
 [639] 1046.29  471.6    -26.38   -247.17  -176.48  -114.11  147.83   147.83   1452.58  614.19
147.83
 [650] -144.89  -2348.96 -36.47   -65.77   185.66   817.51   -26.38   -33.03   397.17   217.35
1016.26
 [661] -48.5    2489.78  539.66   235.84   -16.06   714.83   152.99   579.54   117.55   -174.19
-36.47
 [672] 456.21   494.03   -114.11  215.17   -14.13   -3.18    -257.74  -177.79  789.19   152.99
-33.03
 [683] 258.99   -36.47   407.53   -182.45  -33.03   210.1    418.74   1192.77  -33.03   169.44
606.01
 [694] -70.82   -65.77   8.88     -112.87  228.74   8.88     152.99   -150.6   452.02   550.79
675.77
 [705] -14.13   -33.03   303.47   -33.03   -33.03   -33.03   77.77    -87.53   539.66   -16.06
-14.13
 [716] -114.11  1019.18  -183.9   716.72   667.11   -33.03   415.88   -108.43  -33.03   1239.5
125.89
 [727] 1417.08  -89.68   716.72   -369.2   298.15   415.88   -53.63   -16.06   419.39   390.16
-257.74
 [738] 1017.31  -16.06   582.83   -71.42   1294.2   582.83   147.83   135.42   339.89   369.09
-314.61
 [749] -14.13   176.87   420.84   101.93   -80.33   -96.49   -16.06   379.62   716.72   26.85
314.98
 [760] -33.03   -16.06   246.6    147.83   1468.89  -202.42  343.26   343.26   415.88   714.83
817.51
 [771] 544.83   425.23   154.34   -292.13  -16.06   1085.43  228.26   343.26   197.81   452.02
-36.23
 [782] 305.14   197.81   -174.33  2654.26  714.83   -177.79  147.83   499.63   -16.06   342.78
323.3
 [793] 497.69   466.55   -114.11  -14.13   -114.11  415.88   1651.51  246.6    264.16   1723.15
147.83
 [804] -16.06   -182.45  -71.42   2538.43  147.83   -70.82   -98.76   184.25   -33.03   -14.13
1053.98
 [815] -26.38   147.83   1834.35  2358.44  169.44   675.77   -14.13   147.83   203.41   1605.71
-33.03
 [826] -16.06   -33.03   -16.06   373.58   671.7    1074.27  147.83   -14.13   1486.48  4545.09
-16.06
 [837] 244.11   359.98   -16.06   147.83   1401.39  -16.06   121.98   124.45   1408.87  -179.73
-112.87
 [848] -33.03   -16.06   214.98   -485.23  369.09   314.98   765.84   -14.13   -177.79  147.83
147.83
```

```
 [859] 420.84   -16.06   555.34   -87.53   -26.38   169.44   231.57   -498.47  841.56   961.91
2046.66
 [870] 951.23   431.79   -377.31  152.99   340.29   -16.06   1037.54  379.62   -98.76   606.01
147.83
 [881] -98.76   -65.77   288.17   303.47   -16.06   -89.68   228.26   147.83   147.83   369.09
107.42
 [892] 253.01   -33.03   -33.03   -16.06   -16.06   671.62   -144.89  -150.6   511.47   452.02
1867.57
 [903] 246.6    197.81   849.48   123.67   -82.41   -182.45  -16.06   -80.33   246.6    561.17
-36.47
 [914] -87.53   1.42     -14.13   -3.18    -114.11  77.77    147.83   343.26   147.83   -202.42
-98.76
 [925] 817.51   483.2    -78.31   -16.06   415.88   -78.31   -98.76   -16.06   -114.11  1016.26
-16.06
 [936] 817.51   228.26   -16.06   147.83   1080.66  152.99   -33.03   147.83   234.76   1353.38
425.23
 [947] 419.39   152.99   582.83   369.09   -16.06   -57.55   -33.03   -16.06   234.76   303.47
-80.33
 [958] -16.06   555.34   369.09   -257.74  -33.03   -16.06   -16.06   210.15   737.85   -66.31
3693.11
 [969] -36.47   1304.12  778.22   194.29   147.83   -14.13   147.83   -114.11  2121.9   368.45
-16.06
 [980] 2577.98  952.96   340.29   -71.42   -80.33   -33.03   -98.76   -174.19  425.16   -64.11
848.55
 [991] -26.38   -36.47   419.39   125.89   142.63   -98.76   125.89   797.31   147.83   368.45
 [ reached getOption("max.print") -- omitted 1896 entries ]
10875 Levels: -11522.96 -10971.98 -10103.68 -9612.99 -8424.62 -7334.45 -7092.73 -5076.7 -3700.27
... 19446.88
```

Hide

```
mean(lda_pred$class==test$Jobs_Gross_Margin)
```

```
[1] 0.0003453039
```

Hide

```
Jobs <- train[,1]
ldaModel <- lda(Jobs ~ ., data = train[,2:4])
trainDataReduced <- predict(ldaModel, train[,2:4])$x
testDataReduced <- predict(ldaModel, test[,2:4])$x
```

Linear Regression LDA

Hide

```
#Using the reduced data to evaluate linear regression
trainDataReduced = data.frame(trainDataReduced)
lm1 <- lm(Jobs ~ ., data = trainDataReduced) # builds linear regression model
summary(lm1) # shows the linear regression model summary
```

```

Call:
lm(formula = Jobs ~ ., data = trainDataReduced)

Residuals:
    Min      1Q  Median      3Q     Max 
-8255.9   -50.5    -9.3    56.1 11383.4 

Coefficients:
            Estimate Std. Error  t value Pr(>|t|)    
(Intercept) 296.4176     2.3550  125.870   <2e-16 ***
LD1         -58.4957     0.1937 -301.919   <2e-16 ***
LD2          25.6897     0.6894   37.265   <2e-16 ***
LD3          -3.6201     1.1093   -3.263   0.0011 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 253.5 on 11579 degrees of freedom
Multiple R-squared:  0.8888,    Adjusted R-squared:  0.8888 
F-statistic: 3.085e+04 on 3 and 11579 DF,  p-value: < 2.2e-16
```

Evaluate the model

Hide

```
# Evaluate
testDataReduced <- data.frame(testDataReduced)
pred <- predict(lm1, newdata = testDataReduced)
cor_lm <- cor(pred, test$Jobs_Gross_Margin)
mse_lm <- mean((pred - test$Jobs_Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_lm))
```

```
[1] "cor: 0.959206463271685"
```

Hide

```
print(paste('mse:', mse_lm))
```

```
[1] "mse: 36101.9618087528"
```

Hide

```
print(paste('rmse:', sqrt(mse_lm)))
```

```
[1] "rmse: 190.005162584475"
```

The results for the LDA-reduced dataset for linear regression are shown. These values are the same as values for linear regression on the original dataset. This could be because the dataset does not have high dimensionality so LDA does not have a big impact.

KNN regression for LDA

Hide

```
#Scale data
train_scaled <- trainDataReduced
means <- sapply(train_scaled, mean)
stdvs <- sapply(train_scaled, sd)
train_scaled <- scale(train_scaled, center = means, scale = stdvs)
test_scaled <- scale(testDataReduced, center = means, scale = stdvs)

fit <- knnreg(train_scaled, train$Jobs_Gross_Margin, k = 17)
# Evaluate
pred3 <- predict(fit, test_scaled)
cor_knn2 <- cor(pred3, test$Jobs_Gross_Margin)
mse_knn2 <- mean((pred3 - test$Jobs_Gross_Margin)^2)
print(paste('cor:', cor_knn2))
```

```
[1] "cor: 0.951451843240836"
```

Hide

```
print(paste('mse:', mse_knn2))
```

```
[1] "mse: 43584.7951245781"
```

Hide

```
print(paste('rmse:', sqrt(mse_knn2)))
```

```
[1] "rmse: 208.769717930015"
```

Some accuracy was lost compared to the original data. The correlation, mse, and rsme of the LDA reduced data are shown. The data with all three features had a correlation of 0.95719, a mse of 38144.500667, and a rsme of 195.30617.

Decision Tree

Hide

```
# Download libraries
library(tree)
library(MASS)
# Build tree
tree1 <- tree(train$Jobs_Gross_Margin~., data=trainDataReduced)
summary(tree1)
```

```

Regression tree:
tree(formula = train$Jobs_Gross_Margin ~ ., data = trainDataReduced)
Variables actually used in tree construction:
[1] "LD1"
Number of terminal nodes:  9
Residual mean deviance:  78750 = 911500000 / 11570
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-7069.00   -80.87   -10.79     0.00    94.26  9442.00
```

Hide

```
# Evaluate tree
pred <- predict(tree1, newdata=testDataReduced)
cor_tree <- cor(pred, test$Jobs_Gross_Margin)
mse_tree <- mean((pred-test$Jobs_Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_tree))
```

```
[1] "cor: 0.929033105575348"
```

Hide

```
print(paste('mse:', mse_tree))
```

```
[1] "mse: 61299.5443734591"
```

Hide

```
print(paste("rmse :", sqrt(mse_tree)))
```

```
[1] "rmse : 247.587447931956"
```

The LDA reduced data has greater accuracy than the origial data. The data with all three features had a correlation of 0.914753236805949, a mse of 72983.9813219516, and a rsme of 270.155476202041. ### Pruning

Hide

```
tree_pruned <- prune.tree(tree1, best = 5)
# Evaluate
pred_pruned <- predict(tree_pruned, newdata=testDataReduced)
cor_pruned <- cor(pred_pruned, test$Jobs_Gross_Margin)
mse_pruned <- mean((pred_pruned-test$Jobs_Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_pruned))
```

```
[1] "cor: 0.870229572841003"
```

Hide

```
print(paste('mse:', mse_pruned))
```

```
[1] "mse: 108431.963627748"
```

Hide

```
print(paste('rmse:', sqrt(mse_pruned)))
```

```
[1] "rmse: 329.290090388016"
```

The results of the LDA-reduced dataset after pruning are shown. Some accuracy was lost. The original data received a cor of 0.859246727417677, a mse of 116650.137250333, and a rsme of 341.54082808697 after pruning.

Random Forest

Hide

```
library(randomForest)
set.seed(1234)
rf <- randomForest(train$Jobs_Gross_Margin~., data = trainDataReduced, importance = TRUE)
rf
```

```

Call:
 randomForest(formula = train$Jobs_Gross_Margin ~ ., data = trainDataReduced,      importance =
TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 79489.57
                    % Var explained: 86.24
```

Hide

```
# Evaluate
pred_rf <- predict(rf, newdata = testDataReduced)
cor_rf <- cor(pred_rf, test$Jobs_Gross_Margin)
mse_rf <- mean((pred_rf-test$Jobs_Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_rf))
```

```
[1] "cor: 0.951813212039497"
```

Hide

```
print(paste('mse:', mse_rf))
```

```
[1] "mse: 42101.0163157424"
```

Hide

```
print(paste('rmse:', sqrt(mse_rf)))
```

```
[1] "rmse: 205.185321881811"
```

The results for the LDA-reduced data for random forest are shown. Some accuracy was lost. The original data received a cor of 0.956297620429714, a mse of 38520.938978973, and a rsme of 196.267518909709.

Bagging

Hide

```
bag <- randomForest(train$Jobs_Gross_Margin~., data=trainDataReduced, mtry = 3)
bag
```

```

Call:
 randomForest(formula = train$Jobs_Gross_Margin ~ ., data = trainDataReduced,      mtry = 3)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 79746.23
                    % Var explained: 86.19
```

Hide

```
# Evaluate
pred_bag <- predict(bag, newdata=testDataReduced)
cor_bag <- cor(pred_bag, test$Jobs_Gross_Margin)
mse_bag <- mean((pred_bag-test$Jobs_Gross_Margin)^2)
# Print out values
print(paste('cor:', cor_bag))
```

```
[1] "cor: 0.958144428037841"
```

Hide

```
print(paste('mse:', mse_bag))
```

```
[1] "mse: 36577.5199978943"
```

Hide

```
print(paste('rmse:', sqrt(mse_bag)))
```

```
[1] "rmse: 191.252503246086"
```

The accuracy for the LDA-reduced data is shown. Some accuracy was gained. The original data had an cor of 0.957401702050157, a mse of 37216.2941883724, and a rsme of 192.915251310964 for bagging.

Final Thoughts

Although the PCA and LDA reduced data tend to lose accuracy compared to the original, the metrics were very close and LDA and PCA would be a good way to lower the dimensions of the data if neccessary.

.