California Polytechnic State University

Fall 2024

Analyzing Technology Sales Representatives: The Impact of Traits on Employee Metrics

Evan Murphy, Alex Uchimura, Hunter Blinkenberg, Levi Johansen

GSB 530 Group Project

Professor Chen

December 8, 2024

Introduction

In today's competitive business landscape, leveraging big data has become integral to informed decision-making, enabling companies to stay ahead of the curve and foster sustained growth. Our dataset contains information on 21,990 tech sales representatives for hardware and software teams of a large technology company. It captures performance metrics, peer feedback, customer satisfaction indicators, and other identifying factors, offering a comprehensive view of the traits and skills that define top-performing sales representatives. This data provides a strong foundation for optimizing peer feedback and enhancing customer satisfaction. Complete information on the individual variables is available in appendix D.1.

By analyzing the performance of tech sales employees, we can gain valuable insights into the skills, behaviors, and metrics that drive success in this field. With this data, we can assist tech companies in refining their approaches to hiring, promotion, and customer satisfaction by analyzing key predictors that contribute to higher Net Promoter Scores (NPS) and feedback enabling us to define the characteristics of an ideal tech sales representative. This includes identifying the traits, behaviors, and skills that foster stronger customer relationships and drive higher satisfaction levels. By leveraging these findings, the business can attract, develop, and retain employees who not only perform effectively but also contribute to long-term customer loyalty and business success. Through multiple data analysis methods, we hope to answer the following questions:

I. What is the ideal tech sales rep?

II. What are the best explanatory variables to predict feedback and NPS scores?

III. How can the data assist in salary negotiations?

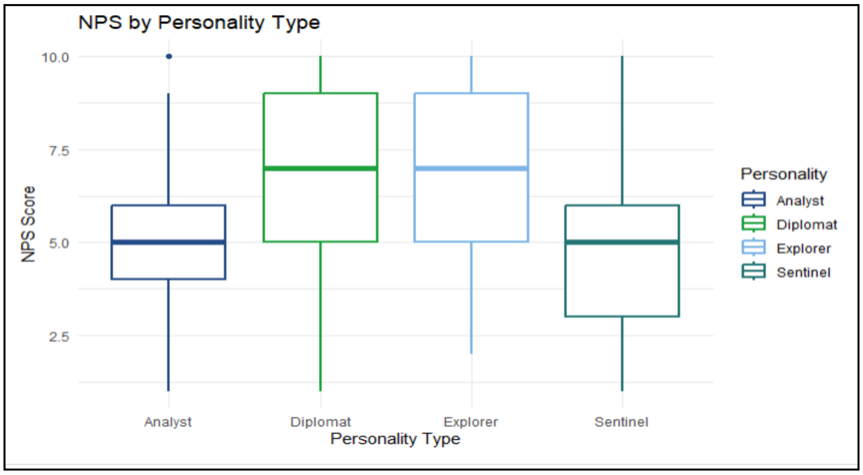
In order to conduct these analyses, the following methods will be employed:

- Supervised Learning:
    - Linear Regression and Regression Tree
- Unsupervised Learning:
    - K-Means Analysis and Principal Component Analysis (PCA)

Using the results of these analyses we can provide recommendations on ideal employee placement, salaries, certification sponsoring, and other metrics in order to drive high peer and customer feedback.

I. What is the Ideal Tech Sales Rep?

When hiring new employees, the diplomat and explorer personality types stand out for enhancing customer satisfaction. These types generate a 38% increase in average NPS compared to their counterparts. We find that diplomats excel in building strong personal connections, while explorers are quick thinkers. These are helpful traits in the sales field, where relationship building is essential. Thus, it would be wise for companies to prioritize these personality types in future hiring decisions, as customers highly appreciate these qualities.

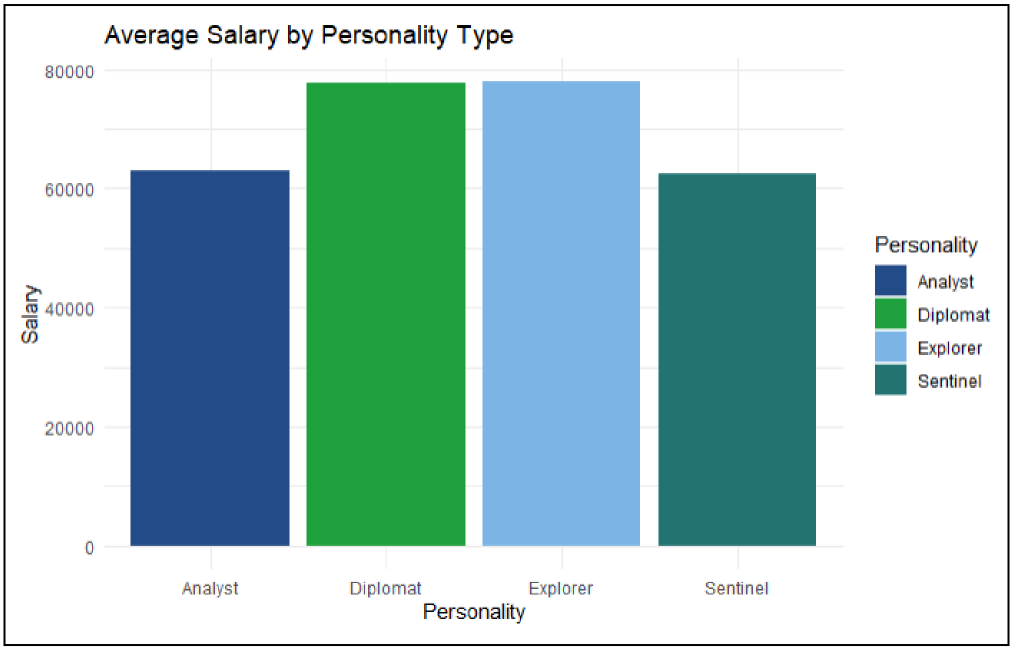


(Appendix A.1)

	Analyst	Diplomat	Explorer	Sentinel
NPS	4.9	6.7	6.7	4.9

(Appendix A.2)

This trend is consistent across both hardware and software sales, so there is no need to align certain personality types with each division. Diplomats and explorers are customers' favorite in both software and hardware sales. However, this improvement in NPS does come at a cost, with the average diplomat and explorer salary being around 24% higher (see visual below and appendix A.3). Despite the salary increase, the impact on customer satisfaction may be worthwhile. Unfortunately, the higher salary associated with diplomat and explorer personality types remains consistent, so there is no cost advantage in hiring one over the other. Both types have higher than-average compensation, reflected by their impact on customer satisfaction.



(Appendix A.4)

However, investing more in a diplomat or explorer does not translate to improved outcomes in the area of feedback within the company. Despite their strong customer satisfaction, these personality types do not show an increase in the internal feedback processes. Additional investment in these areas may not enhance the company's internal environment.

	Analyst	Diplomat	Explorer	Sentinel
Feedback	2.6	2.6	2.6	2.6

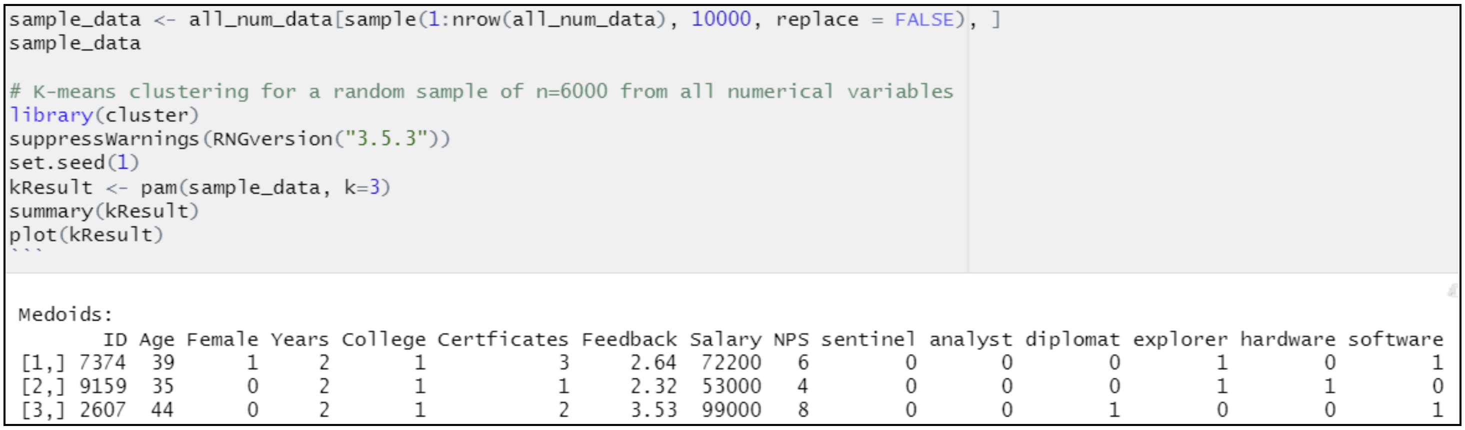
Appendix (A.5)

Since feedback within the groups is consistent, the key question is whether the increase in customer satisfaction that results from these changes will ultimately outweigh the additional costs incurred. Improvements could lead to long-term benefits such as higher customer retention, positive word-of-mouth, and stronger brand loyalty. While further analysis would be required, it is clear that diplomats and explorers provide high levels of value to customers and average levels of feedback, making them great next hires.

II. What are the best explanatory variables to predict feedback and NPS scores?

To discover which variables might best predict feedback and NPS scores, unsupervised K-means cluster analysis was conducted to find cluster characteristics associated with NPS and feedback scores. The sales rep data was cleaned to create a subset of converted categorical variables (Personality, College, and Business) into factor variables and numerical variables to carry out the K-means cluster analysis (see Appendix B.1). Hierarchical clustering was attempted to find an optimal value of K, but was difficult to carry out due to the size of the dataset and omission of categorical variables.

In finding the optimal number of clusters for analysis, multiple values of K were chosen on a subset of 10,000 random observations for computation sake. A K of 3 clusters had the



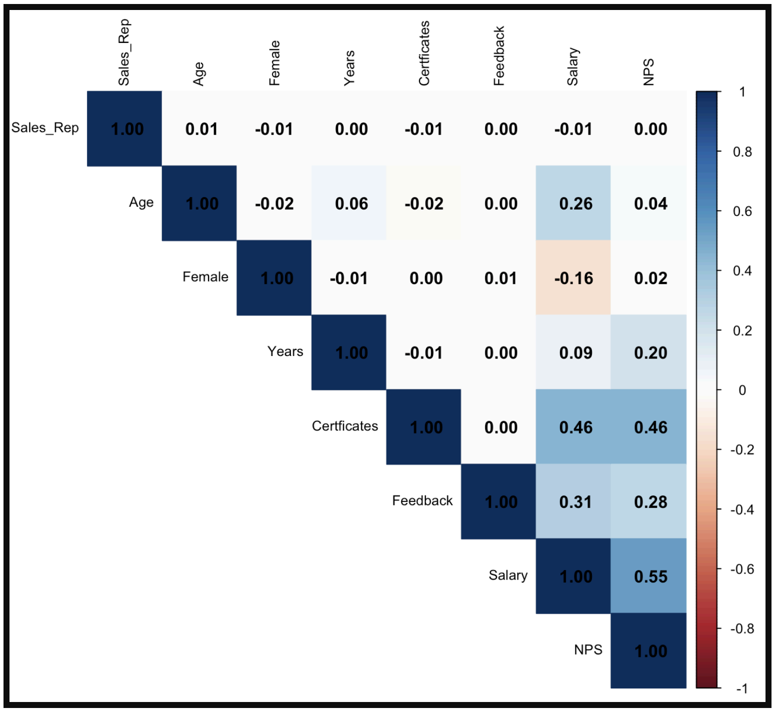
```
sample_data <- all_num_data[sample(1:nrow(all_num_data), 10000, replace = FALSE), ]
sample_data

# K-means clustering for a random sample of n=6000 from all numerical variables
library(cluster)
suppressWarnings(RNGversion("3.5.3"))
set.seed(1)
kResult <- pam(sample_data, k=3)
summary(kResult)
plot(kResult)
```

```
 Medoids:
        ID Age Female Years College Certficates Feedback Salary NPS sentinel analyst diplomat explorer hardware software
 [1,] 7374  39      1     2       1           3     2.64  72200   6        0       0        0        1        0        1
 [2,] 9159  35      0     2       1           1     2.32  53000   4        0       0        0        1        1        0
 [3,] 2607  44      0     2       1           2     3.53  99000   8        0       0        1        0        0        1
```

highest average silhouette width of the total data set of 0.5146, compared to a K of 2, 4, 5, and 6. From the K-means cluster summary above, we can see that NPS and feedback score levels are associated with specific medoids in the cluster analysis. Cluster 3 has the highest NPS and feedback scores (8 and 3.53, respectively) with diplomat personality type, software, male, an above average salary, and having gone to college as primary predictors. Medoids can be used to predict NPS and feedback scores from each given cluster since they are the most central cluster point. Cluster 2 has the lowest feedback and NPS scores which can be associated with female explorers who work in hardware with fewer certificates and have a lower salary. These were the starkest differences between the two most opposing clusters. Age was omitted, as both clusters were somewhat close in age. These results can be shown in the table below.

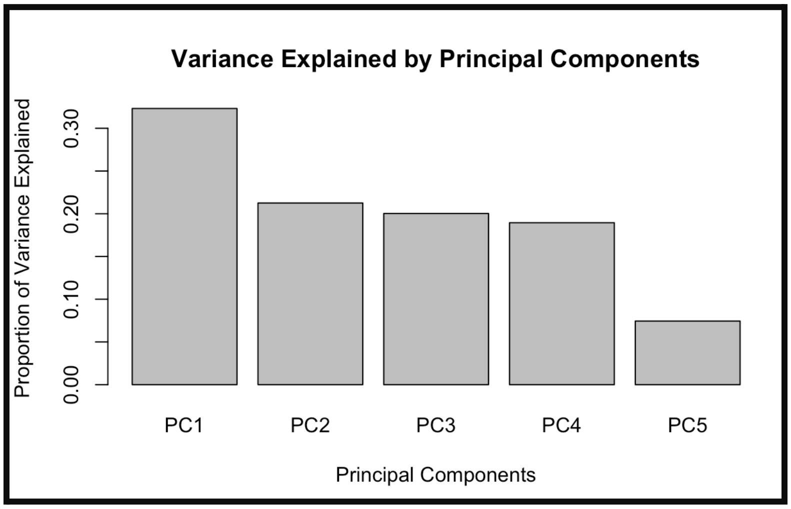
Predictors	**High NPS and Feedback**	**Low NPS and Feedback**
Personality	Diplomat	Explorer
Salary	Above average	Below average
Business	Software	Hardware
College	Yes	Yes
Female	No	Yes
Certificates	2	1

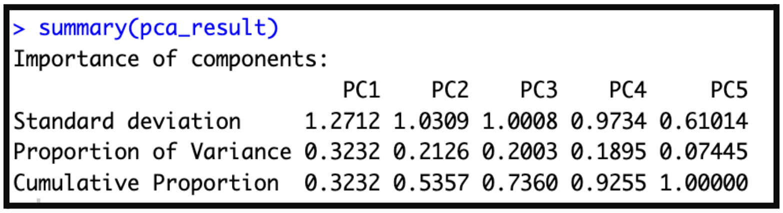
Additionally, PCA was conducted to identify the best variables for predicting NPS and Feedback, but several issues occurred. First, the initial number of variables (10) is relatively small, making model fitting straightforward, thus reducing the utility of PCA. Additionally, PCA is designed for continuous numerical data, but only six of the variables in the dataset are numerical. Among these, Certificates range from 1 to 6, and Years range from 1 to 13, making them more akin to ordinal data rather than continuous variables, which raises questions about their suitability for PCA. See Appendix B.2. Furthermore, PCA is most useful when there is a high degree of correlation among predictor variables. In this case, the correlation matrix reveals minimal correlation, with the highest correlation coefficient being only 0.55. Given the lack of substantial correlation among the variables, the potential benefit of using PCA is significantly limited in this analysis.



Appendix B.3

The PCA conducted on the six numerical variables resulted in five principal components. The first four PCs explained 90% of the variance in the data. Using these four principal components to predict NPS with a linear regression model yielded an Adjusted R-squared of 0.41. The cross-validated Mean Squared Error (MSE) was 2.71, reflecting the prediction error on unseen data. While PCA effectively reduced the dimensionality, the resulting model has a slightly lower explanatory power and higher error compared to models using the original predictors.





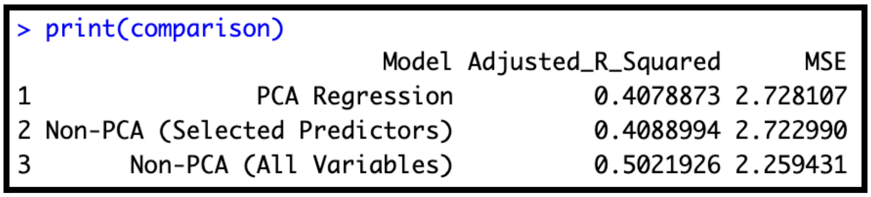
```
> summary(pca_result)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5
Standard deviation     1.2712 1.0309 1.0008 0.9734 0.61014
Proportion of Variance 0.3232 0.2126 0.2003 0.1895 0.07445
Cumulative Proportion  0.3232 0.5357 0.7360 0.9255 1.00000
```

Appendix B.4

Appendix B.5

When using linear regression with only the same six numerical variables, the model achieved slightly better results, with an Adjusted R-squared of 0.408 and a cross-validated Mean Squared Error (MSE) of 2.722. This indicates that PCA does not improve the analysis, as the original variables perform better without dimensionality reduction.

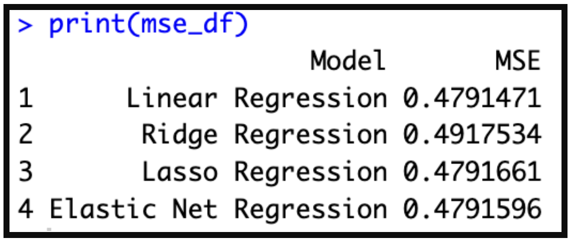
When using a linear regression model to predict NPS using all predictors, the model explains approximately 50.59% of the variance in NPS, with an Adjusted R² of 0.5059. This demonstrates reasonable predictive accuracy with a Mean Squared Error (MSE) of 2.259 on test data. This further demonstrates the PCA is not beneficial to our analysis.



```
> print(comparison)
                            Model Adjusted_R_Squared      MSE
1                  PCA Regression          0.4078873 2.728107
2 Non-PCA (Selected Predictors)            0.4088994 2.722990
3          Non-PCA (All Variables)         0.5021926 2.259431
```

Appendix B.6

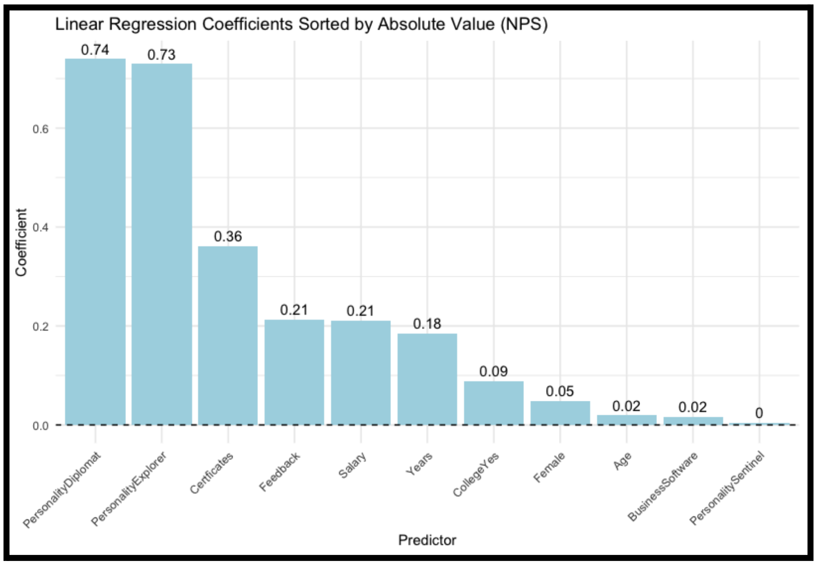
When using regressions to find the most important variables to predict NPS we want to expand away from just linear and also consider ridge, lasso, and elastic net. Here, we use dummy variables for categorical variables and standardize numerical variables so we can see the effect of each variable on the same scale. Then used training and validation data to train the models. When predicting NPS we find that our best model based on cross validated MSE is still linear regression.



```
> print(mse_df)
                    Model       MSE
1       Linear Regression 0.4791471
2        Ridge Regression 0.4917534
3        Lasso Regression 0.4791661
4 Elastic Net Regression 0.4791596
```

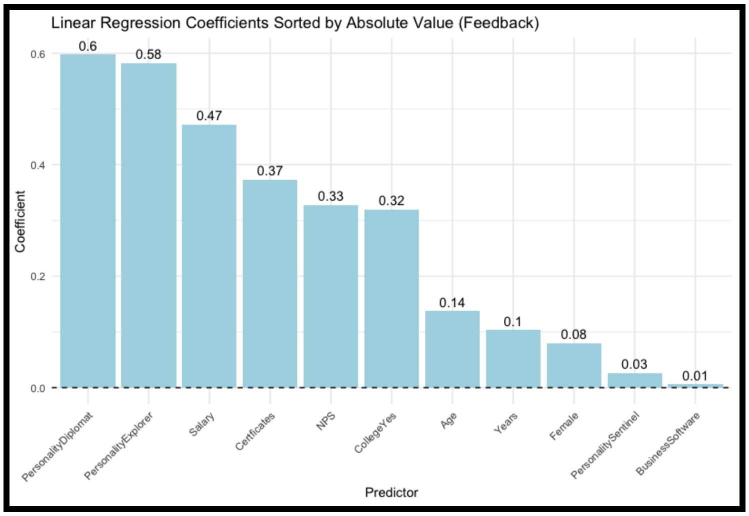
Appendix B.7

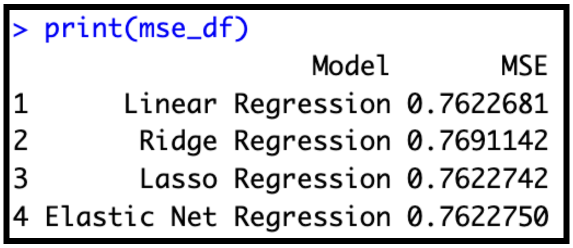
Based on our best model, Linear Regression, The most important variables for predicting NPS are Diplomat, Explorer, and Certificates.



Appendix B.8

When we do this analysis with Feedback as the dependent variable we see a very similar story with linear regression still being the best model and the most important variables being Diplomat, Explorer, and Salary.





```
> print(mse_df)
                  Model       MSE
1     Linear Regression 0.7622681
2      Ridge Regression 0.7691142
3      Lasso Regression 0.7622742
4 Elastic Net Regression 0.7622750
```

Appendix B.7

Appendix B.8

What we notice when comparing the two models is that personality types Diplomat and Explorer have a stronger influence on NPS while other factors like Salary and Certificates have an increased role in predicting Feedback.

III. How can the data assist in salary negotiations?

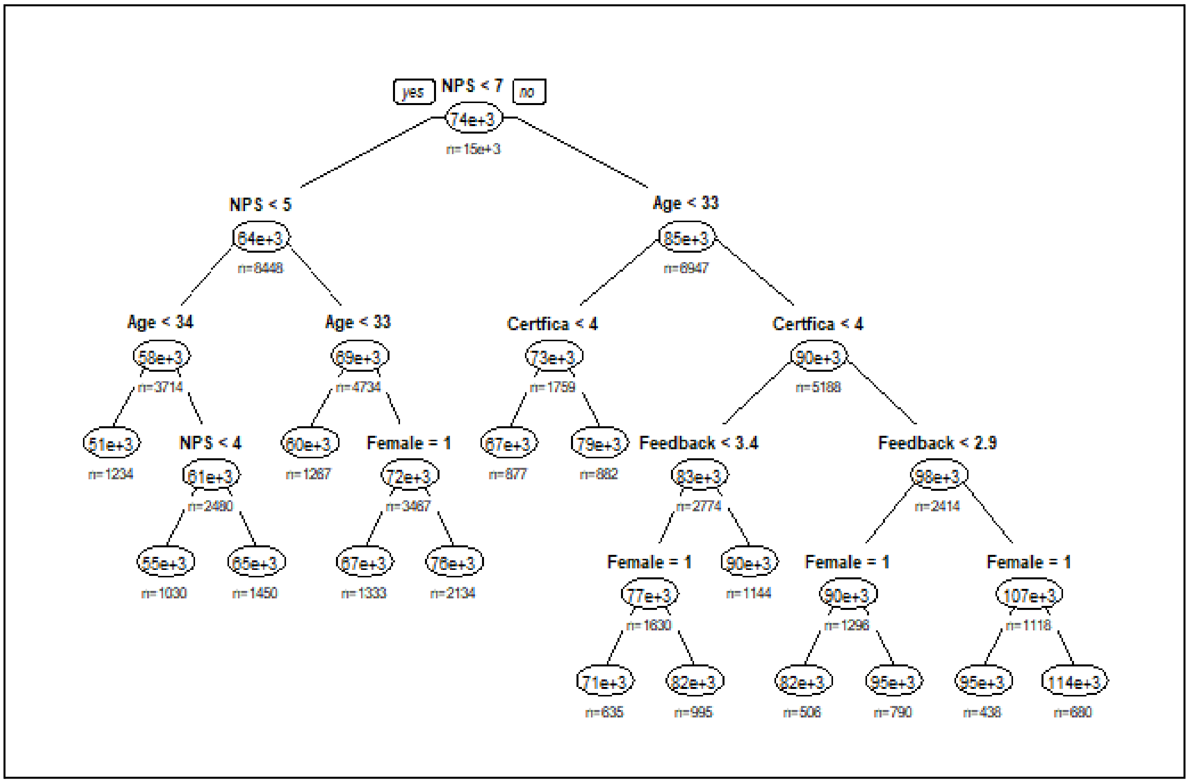
Providing employees with fair and deserved salaries is an integral part of business. Maintaining employee satisfaction and paying them fairly for the amount and caliber of their work must be balanced with ensuring that the business still has room to expand and reinvest into new ventures. In order to enhance clarity and maintain brevity, this analysis utilizes a decision tree. It shows one node at a time, and flows downward until a final salary node is reached. This makes it easy for recruiters, hiring managers, or anyone who is in the process of hiring or negotiating a new salary to read the chart and come to a starting point on where the negotiations should begin.

As a result of this decision tree, the main factor when deciding salary is NPS. Beyond this, age, number of certificates, feedback, and gender are all major predictors too. According to the tree, salaries range from $51,000 all the way up to $114,000. There are fifteen leaf nodes on the tree which suggests an adequate amount of salary options. An issue we encountered with this tree and the size of the dataset was pruning it enough so that the tree was still viable to make good decisions, but not be too big to read. Balancing these two parameters left us with a complexity parameter of .0057 and seemed to yield the best results for analysis. For judging error, the metric of choice was MPE, or mean percentage error. When testing the decision tree on test data, the model displayed an MPE of -5.1% (see appendix C.2), meaning the tree consistently underestimates salary by about 5%, which is a low percentage error. This helps firms begin negotiations at a slightly lower amount than what they would be willing to pay.

While this tree would be a great start to create a baseline for salary for new hires or promotions, we understand that it is an estimate based on current employees, not a one-size-fits-all solution. As businesses evolve and employee roles diversify, this model should be continually refined and supplemented with contextual insights to ensure alignment with employee expectations and organizational goals. Overall, it is imperative to analyze all cases of salary negotiation/adjustment individually and holistically in accordance with business needs and budget.

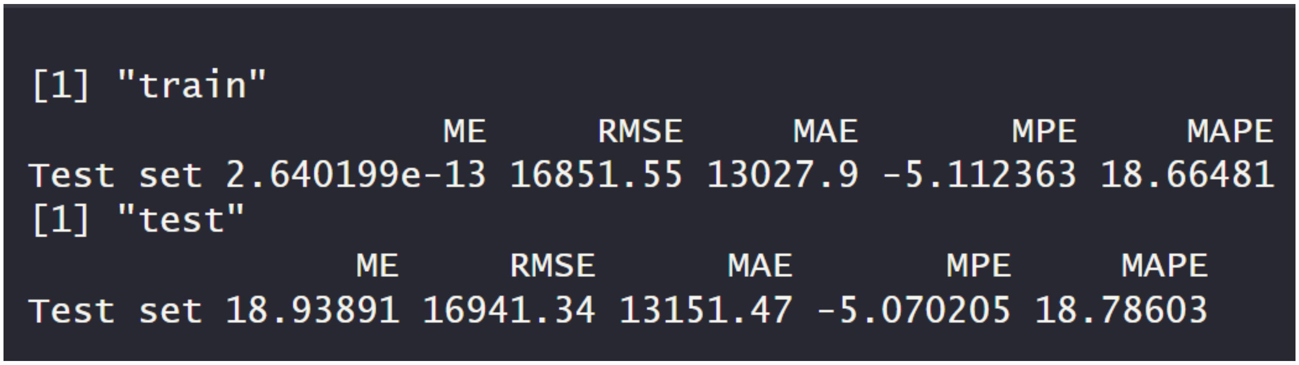
One major reason that this tree should simply be a reference is the presence of the "female" leaves in the tree. According to the data, females consistently have a lower salary holding all other characteristics equal to each other. For example, male sales reps with an NPS higher than 7, are over 33 years old, possess four or more certificates, and have a feedback score of over 2.9 garner a salary of $114,000 on average. For a female with those same characteristics, the predicted salary is only $95,000. This data exposes a gender pay gap that must be addressed when negotiating salaries moving forward, and further emphasizes more scrutiny than simply relying on the tree to make final decisions.

Decision Tree balancing Visibility and Accuracy



Appendix C.1

Train/Test Error for Decision Tree



```
[1] "train"
                     ME     RMSE     MAE       MPE     MAPE
Test set 2.640199e-13 16851.55 13027.9 -5.112363 18.66481
[1] "test"
                ME     RMSE      MAE       MPE     MAPE
Test set 18.93891 16941.34 13151.47 -5.070205 18.78603
```

Appendix C.2

Conclusion

Overall, this dataset offers three useful insights into 3 main questions: What is the ideal tech sales rep? What are the best explanatory variables to predict feedback and NPS scores? How can the data assist in salary negotiations?

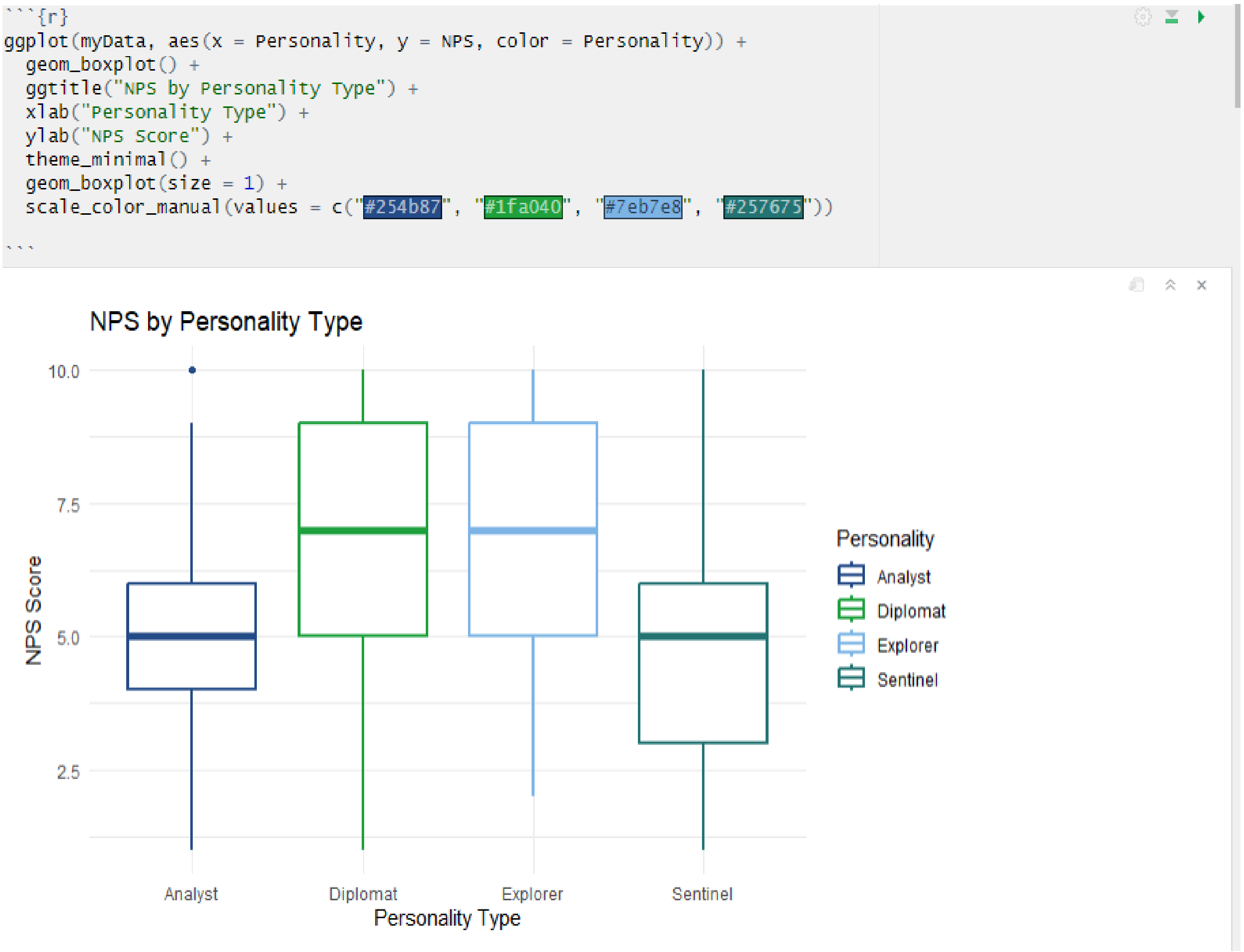
Firstly, the ideal tech sales rep can be defined as having higher NPS and feedback scores. We found that being a diplomat or explorer personality type is highly correlated with higher score metrics through descriptive statistics from part I. Intuitively, these personality types are more likely to work well with people and be more engaging overall compared to analysts and sentinels. However, diplomats and explorers are also associated with higher salaries, so a firm should decide whether they value better customer satisfaction and workplace scores at the cost of hiring more higher paid personality types.

When it comes to predicting primary explanatory variables for NPS and feedback scores, K-means clustering, PCA, and linear regression were performed. The 3 clusters from the K-means analysis found that clusters with higher NPS and feedback scores had medoids around college educated, diplomats who work in software with above average salaries and number of certificates. These findings align with what was expected earlier with the addition of software and a higher number of certificates playing a role in predicting higher NPS and feedback. Additionally, PCA was carried out, but was found to have less predictive power in a linear regression compared to using the original variables. This was mainly due to a lack of continuous numerical explanatory variables and correlation between original variables. The linear regression found that Diplomat, Explorer and Certificates were most influential in predicting NPS due to high coefficient values, whereas Diplomat, Explorer and Salary were most influential in predicting feedback scores. These findings suggest that firms should prioritize diplomats and explorers when hiring, sponsor certifications for current employees, and provide a healthy workplace environment and balance to ensure employee retention to maximize NPS and feedback scores.

Finally, a regression tree was constructed to illuminate where proper salary negotiations should begin based on several important factors within the company. NPS, Age, Certificates, Feedback, Gender were the biggest predictors in this regard, and predicted salaries with an average of 95% accuracy. This easy to navigate tree provides a great reference point for anyone in the company wishing to use data to predict/assess what a salary should be for a sales rep based on the largest characteristics.

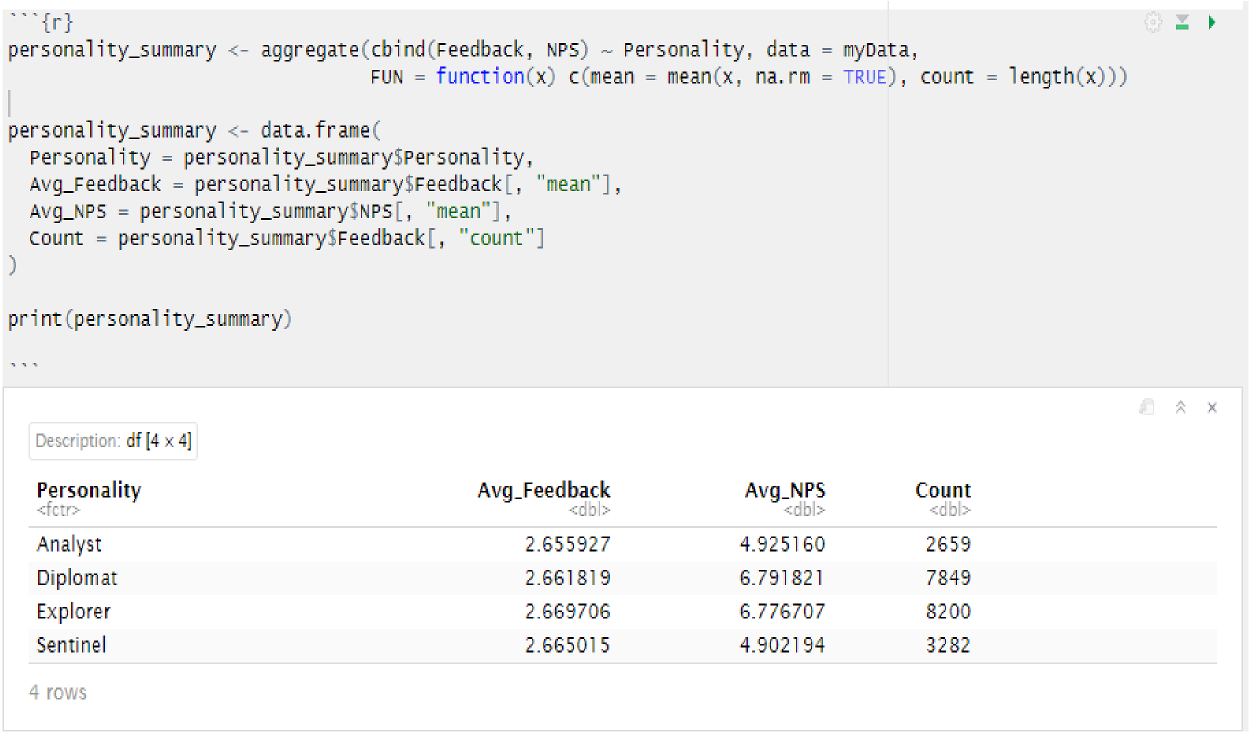
Appendix

A.1



~~~
```{r}
ggplot(myData, aes(x = Personality, y = NPS, color = Personality)) +
  geom_boxplot() +
  ggtitle("NPS by Personality Type") +
  xlab("Personality Type") +
  ylab("NPS Score") +
  theme_minimal() +
  geom_boxplot(size = 1) +
  scale_color_manual(values = c("#254b87", "#1fa040", "#7eb7e8", "#257675"))

```
~~~

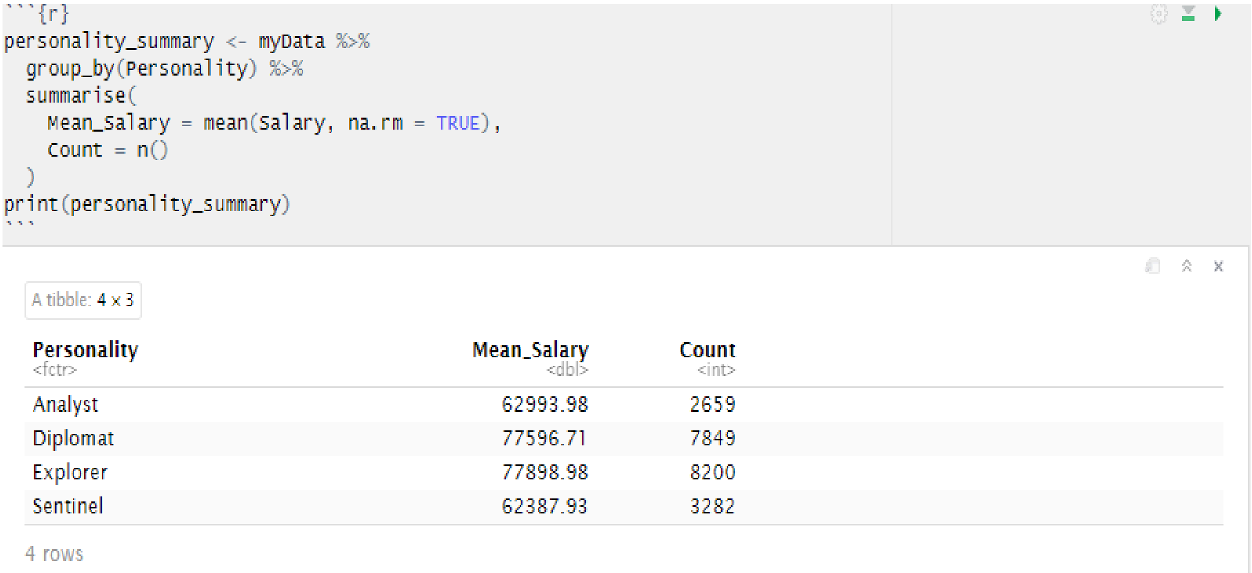
A.2



~~~
```{r}
personality_summary <- aggregate(cbind(Feedback, NPS) ~ Personality, data = myData,
                                 FUN = function(x) c(mean = mean(x, na.rm = TRUE), count = length(x)))

personality_summary <- data.frame(
  Personality = personality_summary$Personality,
  Avg_Feedback = personality_summary$Feedback[, "mean"],
  Avg_NPS = personality_summary$NPS[, "mean"],
  Count = personality_summary$Feedback[, "count"]
)

print(personality_summary)

```
~~~

```
Description: df [4 x 4]

Personality                          Avg_Feedback      Avg_NPS      Count
<fctr>                                      <dbl>        <dbl>      <dbl>
Analyst                                  2.655927     4.925160       2659
Diplomat                                 2.661819     6.791821       7849
Explorer                                 2.669706     6.776707       8200
Sentinel                                 2.665015     4.902194       3282

4 rows
```

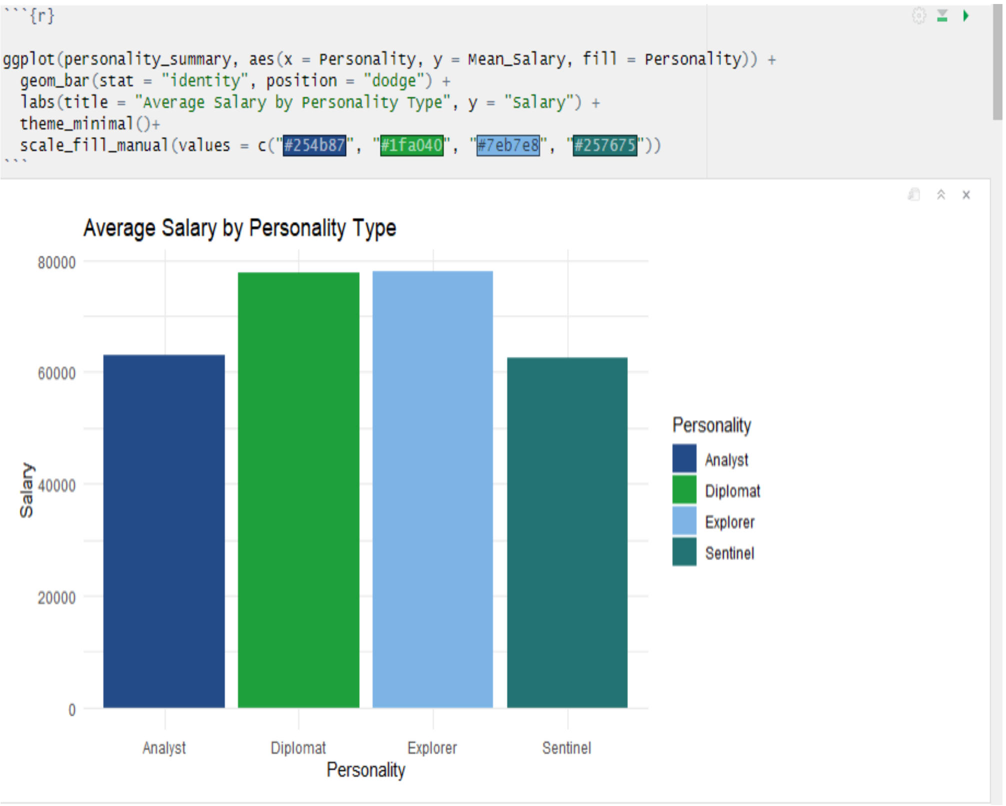
A.3



~~~
```{r}
personality_summary <- myData %>%
  group_by(Personality) %>%
  summarise(
    Mean_Salary = mean(Salary, na.rm = TRUE),
    Count = n()
  )
print(personality_summary)
```
~~~

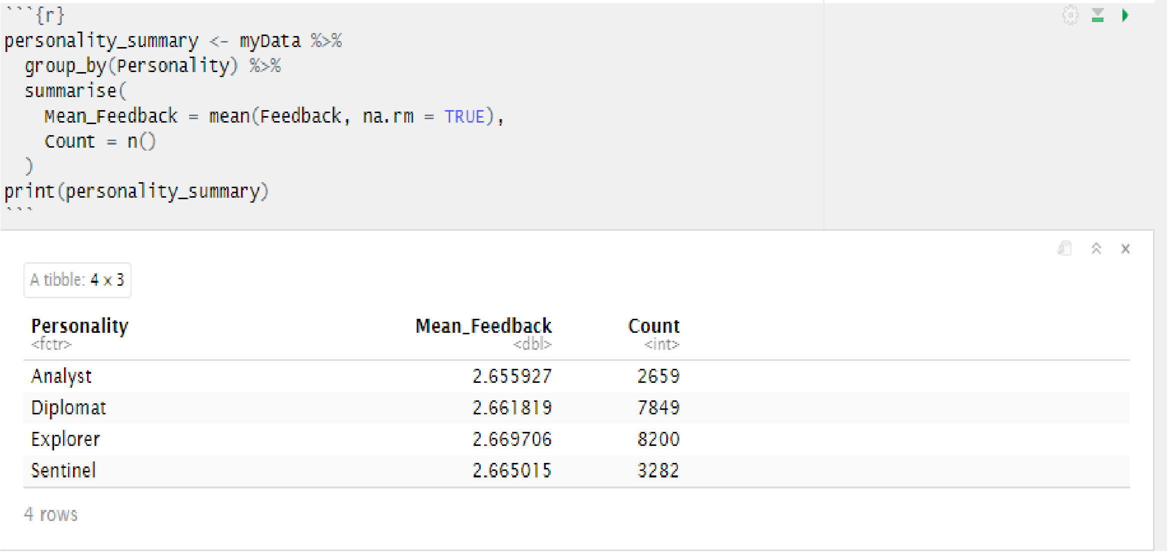
```
A tibble: 4 x 3

Personality                          Mean_Salary      Count
<fctr>                                     <dbl>      <int>
Analyst                                 62993.98       2659
Diplomat                                77596.71       7849
Explorer                                77898.98       8200
Sentinel                                62387.93       3282

4 rows
```

A.4



~~~
```{r}

ggplot(personality_summary, aes(x = Personality, y = Mean_Salary, fill = Personality)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Salary by Personality Type", y = "Salary") +
  theme_minimal()+
  scale_fill_manual(values = c("#254b87", "#1fa040", "#7eb7e8", "#257675"))
```
~~~

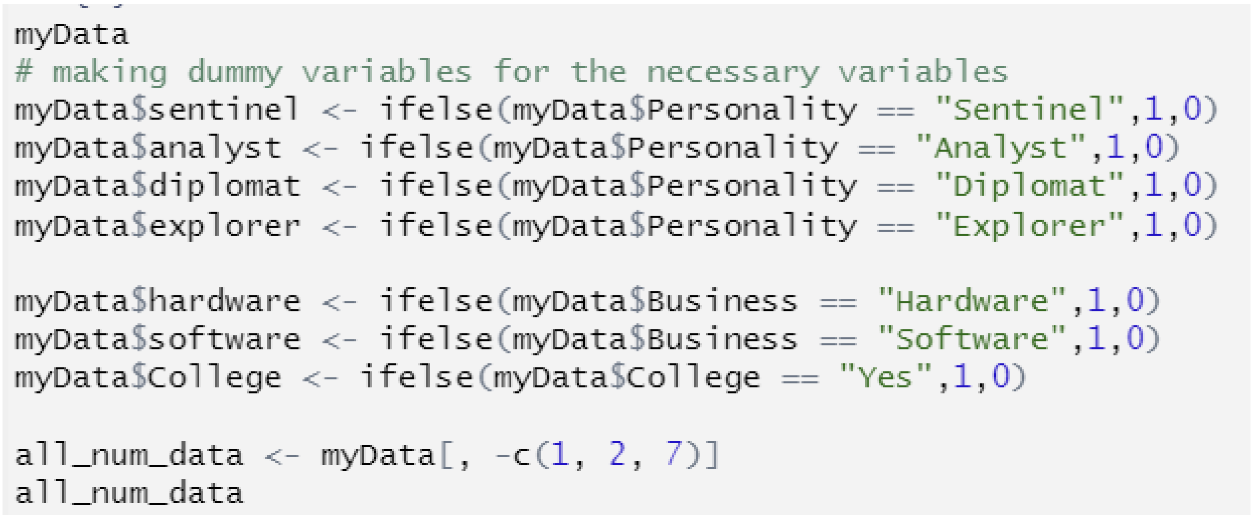
A.5



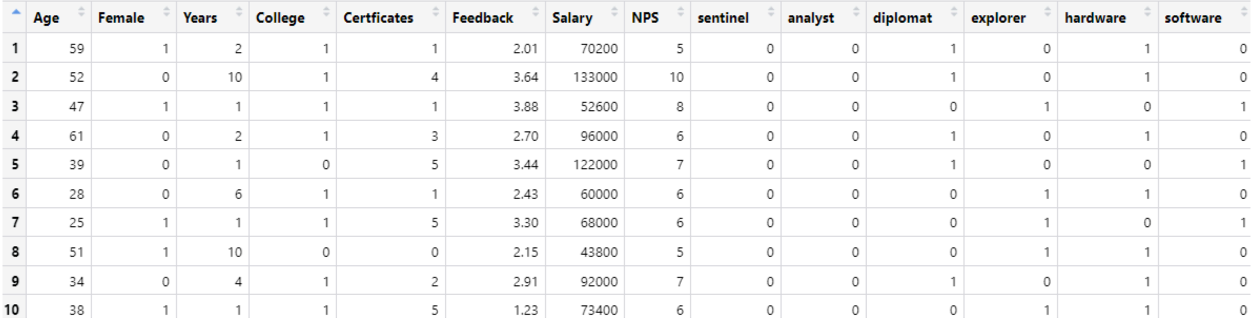
~~~
```{r}
personality_summary <- myData %>%
  group_by(Personality) %>%
  summarise(
    Mean_Feedback = mean(Feedback, na.rm = TRUE),
    Count = n()
  )
print(personality_summary)
```
~~~

```
A tibble: 4 x 3

Personality                     Mean_Feedback      Count
<fctr>                                  <dbl>      <int>
Analyst                              2.655927       2659
Diplomat                             2.661819       7849
Explorer                             2.669706       8200
Sentinel                             2.665015       3282

4 rows
```

B.1: Data Cleaning for K-Means Analysis



```
myData
# making dummy variables for the necessary variables
myData$sentinel <- ifelse(myData$Personality == "Sentinel",1,0)
myData$analyst <- ifelse(myData$Personality == "Analyst",1,0)
myData$diplomat <- ifelse(myData$Personality == "Diplomat",1,0)
myData$explorer <- ifelse(myData$Personality == "Explorer",1,0)

myData$hardware <- ifelse(myData$Business == "Hardware",1,0)
myData$software <- ifelse(myData$Business == "Software",1,0)
myData$College <- ifelse(myData$College == "Yes",1,0)

all_num_data <- myData[, -c(1, 2, 7)]
all_num_data
```



	Age	Female	Years	College	Certficates	Feedback	Salary	NPS	sentinel	analyst	diplomat	explorer	hardware	software
1	59	1	2	1	1	2.01	70200	5	0	0	1	0	1	0
2	52	0	10	1	4	3.64	133000	10	0	0	1	0	1	0
3	47	1	1	1	1	3.88	52600	8	0	0	0	1	0	1
4	61	0	2	1	3	2.70	96000	6	0	0	1	0	1	0
5	39	0	1	0	5	3.44	122000	7	0	0	1	0	0	1
6	28	0	6	1	1	2.43	60000	6	0	0	0	1	1	0
7	25	1	1	1	5	3.30	68000	6	0	0	0	1	0	1
8	51	1	10	0	0	2.15	43800	5	0	0	0	1	1	0
9	34	0	4	1	2	2.91	92000	7	0	0	1	0	1	0
10	38	1	1	1	5	1.23	73400	6	0	0	0	1	1	0

B.2: PCA Code

```
library(readxl)
myData <- read_excel("Downloads/Big_Data_Files.xlsx",
                     sheet = "TechSales_Reps")
#### PCA with only numerical variables. ###
head(myData)
pca_data <- myData[, c( "Age", "Years",
                        "Certficates",
                        "Feedback", "Salary")]
# Standardize the data
pca_data_scaled <- scale(pca_data)
head(pca_data_scaled)
# Perform PCA
pca_result <- prcomp(pca_data_scaled, scale. = TRUE)
# Summarize PCA results
summary(pca_result)
```

B.3: Code for Correlation Matrix.

```
### Correlation Matrix ###
# Select only numeric columns
numeric_data <- myData[, sapply(myData, is.numeric)]
# Compute correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")  # Excludes missing values
# Print the correlation matrix
print(cor_matrix)

# Visualize the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.8, tl.col = "black")

# Visualize the correlation matrix with values
corrplot(cor_matrix,
        method = "color",
        type = "upper",
        tl.cex = 0.8,
        tl.col = "black",
        addCoef.col = "black")
```

B.4: Explained Variance Bar Chart:

```
explained_variance <- summary(pca_result)$importance[2, ]
barplot(explained_variance,
       main = "Variance Explained by Principal Components",
       xlab = "Principal Components",
       ylab = "Proportion of Variance Explained")
```

B.5: Summary of PCA results

```
summary(pca_results)
```

B.6: Full code for PCA regression, numerical only regression and all variables regression:

```
# Select predictors for selected regression
predictors <- c("Age", "Years", "Certficates", "Feedback", "Salary")
response <- "NPS"
selected_regression_data <- myData[, c(predictors, response)]

# Full regression data with all variables
full_regression_data <- myData
```

```
### Non-PCA Workflow (Selected Predictors) ###
# Split selected regression data into training and testing sets
set.seed(1)
train_index <- sample(1:nrow(selected_regression_data), size = 0.8 *
nrow(selected_regression_data))
train_data_selected <- selected_regression_data[train_index, ]
test_data_selected <- selected_regression_data[-train_index, ]

# Fit linear regression model without PCA (selected predictors)
model_selected <- lm(NPS ~ ., data = train_data_selected)

# Calculate metrics for non-PCA regression with selected predictors
r_squared_selected <- summary(model_selected)$adj.r.squared
mse_selected <- mean((test_data_selected$NPS - predict(model_selected,
test_data_selected))^2)

### Non-PCA Workflow (All Variables) ###
# Split full regression data into training and testing sets
set.seed(1)
train_index <- sample(1:nrow(full_regression_data), size = 0.8 * nrow(full_regression_data))
train_data_full <- full_regression_data[train_index, ]
test_data_full <- full_regression_data[-train_index, ]

# Fit linear regression model with all variables
model_full <- lm(NPS ~ ., data = train_data_full)

# Calculate metrics for non-PCA regression with all variables
r_squared_full <- summary(model_full)$adj.r.squared
mse_full <- mean((test_data_full$NPS - predict(model_full, test_data_full))^2)

### Comparison Data Frame ###
comparison <- data.frame(
  Model = c("PCA Regression", "Non-PCA Regression (Numeric Predictors)", "Non-PCA
Regression (All Variables)"),
  Adjusted_R_Squared = c(r_squared_pca, r_squared_selected, r_squared_full),
  MSE = c(mse_pca, mse_selected, mse_full)
)
```

B.7: All code for regression analysis. Including, Linear, Ridge, Lasso, and Elastic Net.
Note: Code used both for NPS and Feedback.

```
# Load the Dataset
myData <- read_excel("Downloads/Big_Data_Files.xlsx", sheet = "TechSales_Reps")

# Convert categorical variables to factors
myData$Business <- as.factor(myData$Business)
myData$College <- as.factor(myData$College)
myData$Personality <- as.factor(myData$Personality)

# Define the formula for regression
formula <- NPS ~ Age + Female + Years + Certficates + Feedback + Salary + Business +
College + Personality

##### Data Preprocessing #####

# Split the data into training and testing sets
set.seed(42)
train_index <- sample(1:nrow(myData), size = 0.8 * nrow(myData))
train_data <- myData[train_index, ]
test_data <- myData[-train_index, ]

# Standardize numeric variables
numeric_columns <- sapply(myData, is.numeric)
scaled_train <- as.data.frame(scale(train_data[, numeric_columns]))
scaled_test <- as.data.frame(scale(test_data[, numeric_columns], center = attr(scale(train_data[,
numeric_columns]), "scaled:center"),
                             scale = attr(scale(train_data[, numeric_columns]), "scaled:scale")))
scaled_train$Business <- train_data$Business
scaled_train$College <- train_data$College
scaled_train$Personality <- train_data$Personality
scaled_test$Business <- test_data$Business
scaled_test$College <- test_data$College
scaled_test$Personality <- test_data$Personality

##### Model Training #####

# Create Model Matrices
X_train <- model.matrix(formula, data = scaled_train)[, -1]  # Remove intercept
```

```
y_train <- scaled_train$NPS
X_test <- model.matrix(formula, data = scaled_test)[, -1]
y_test <- scaled_test$NPS

# Linear Regression
linear_model <- lm(formula, data = scaled_train)
linear_predictions <- predict(linear_model, scaled_test)
linear_mse <- mean((y_test - linear_predictions)^2)

# Ridge Regression
ridge_model <- cv.glmnet(X_train, y_train, alpha = 0, standardize = FALSE)
ridge_best_lambda <- ridge_model$lambda.min
ridge_predictions <- predict(ridge_model, s = ridge_best_lambda, newx = X_test)
ridge_mse <- mean((y_test - ridge_predictions)^2)

# Lasso Regression
lasso_model <- cv.glmnet(X_train, y_train, alpha = 1, standardize = FALSE)
lasso_best_lambda <- lasso_model$lambda.min
lasso_predictions <- predict(lasso_model, s = lasso_best_lambda, newx = X_test)
lasso_mse <- mean((y_test - lasso_predictions)^2)

# Elastic Net Regression
elastic_net_model <- cv.glmnet(X_train, y_train, alpha = 0.5, standardize = FALSE)
elastic_net_best_lambda <- elastic_net_model$lambda.min
elastic_net_predictions <- predict(elastic_net_model, s = elastic_net_best_lambda, newx =
X_test)
elastic_net_mse <- mean((y_test - elastic_net_predictions)^2)

##### Combine Coefficients #####

# Extract Coefficients
linear_coefs <- coef(linear_model)
ridge_coefs <- as.vector(coef(ridge_model, s = ridge_best_lambda))
lasso_coefs <- as.vector(coef(lasso_model, s = lasso_best_lambda))
elastic_net_coefs <- as.vector(coef(elastic_net_model, s = elastic_net_best_lambda))

##### Model Performance #####

# Create a DataFrame of MSE values
mse_df <- data.frame(
  Model = c("Linear Regression", "Ridge Regression", "Lasso Regression", "Elastic Net Regression"),
  MSE = c(linear_mse, ridge_mse, lasso_mse, elastic_net_mse)
)

# Display Results
print("Coefficients DataFrame:")
print(coefficients_df)

print("MSE DataFrame:")
print(mse_df)
```

B.8: Variable Significance Visualization.

```
# Plot the coefficients with values displayed on top of each bar
ggplot(linear_coef_df_sorted, aes(x = reorder(Predictor, -Absolute_Coefficient), y = Coefficient)) +
  geom_bar(stat = "identity", fill = "lightblue", alpha = 1) +  # Use lighter blue
  geom_text(aes(label = round(Coefficient, 2)), vjust = ifelse(linear_coef_df_sorted$Coefficient > 0, -0.5, 1.5)) +  # Add bar values
  geom_hline(yintercept = 0, color = "black", linetype = "dashed") +
  labs(
    title = "Linear Regression Coefficients Sorted by Absolute Value (NPS)",
    x = "Predictor",
    y = "Coefficient"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

C.1:

```
set.seed(1)
dataclean$Business <- as.factor(dataclean$Business)
dataclean$Personality <- as.factor(dataclean$Personality)
myIndex <- createDataPartition(dataclean$Salary, p= .7, list = FALSE)
trainSet <- dataclean[myIndex,]
validationSet <- dataclean[-myIndex,]
set.seed(1)
```

```
default_tree <- rpart(Salary ~., data = trainSet, method = "anova")
summary(default_tree)
prp(default_tree, type = 1, extra = 1, under = TRUE)
set.seed(1)
full_tree <- rpart(Salary ~ ., data = trainSet, method = "anova", cp = 0.00001, minsplit = 100,
minbucket = 10)
prp(full_tree, type = 1, extra = 1, under = TRUE)
```

C.2:

```
optimal_cp <- full_tree$cptable[which.min(full_tree$cptable[, "xerror"]), "CP"]
print(optimal_cp)
pruned_tree <- prune(full_tree, cp = 0.0057)
prp(pruned_tree, type = 1, extra = 1, under = TRUE)
predicted_value_f <- predict(pruned_tree, validationSet)
pred_train_f<- predict(pruned_tree, trainSet)
print("train")
accuracy(pred_train_f, trainSet$Salary)
print("test")
accuracy(predicted_value_f, validationSet$Salary)
```

D.1:

TechSales_Reps

The ***TechSales_Reps*** data contain records of 21,990 sales representatives from the hardware and software product groups of a high-tech company. For each employee, the data include socio-demographic and education information, salary, sales performance, and a personality indicator. Also included in the data is the net promoter score, which is an indicator of customer satisfaction with each sales rep.

TABLE A.7 Data Dictionary for Tech Sales Reps Data

Variable name	Description or possible values
Rep	A unique ID for each sales representative
Business	One of the two product groups: Hardware and Software
Age	Employee's age
Female	1 – female 0 – otherwise
Years	The number of years the employee has been employed at the company
College	Whether or not the employee has a four-year college degree (Yes/No)
Personality	Analyst: This personality type exemplifies rationality. Analysts tend to be open-minded and strong-willed. They like to work independently and usually approach things from a very practical perspective. Diplomat: Diplomats care about people and tend to have a lot of empathy toward others. They exemplify cooperation and diplomacy. Explorer: Explorers are highly practical and can think on their feet. They tend to be very good at making quick, rational decisions in difficult situations. Sentinel: Sentinels are cooperative and practical. They like stability, order, and security. People with this personality type tend to be hardworking and meticulous.
Certificates	The number of relevant professional certifications each employee has earned
Feedback	The average feedback score that each employee receives from his or her peers and supervisor on the 360-degree annual evaluation. The possible scores range from 0 (lowest) to 4 (highest).
Salary	Annual base salary of each employee
NPS	The net promoter score (NPS) is a key indicator of customer satisfaction and loyalty.