AD699_Assignment_2(Raka)

2023-10-10

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(carData)
```

```
## Warning: package 'carData' was built under R version 4.2.3
```

```
?Salaries
```

```
## starting httpd help server ... done
```

● Simple Linear Regression: 1.

```
View(Salaries)
```

2.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(Salaries)
```

```
## Rows: 397
## Columns: 6
## $ rank          <fct> Prof, Prof, AsstProf, Prof, Prof, AssocProf, Prof, Prof,…
## $ discipline    <fct> B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, A, A,…
## $ yrs.since.phd <int> 19, 20, 4, 45, 40, 6, 30, 45, 21, 18, 12, 7, 1, 2, 20, 1…
## $ yrs.service   <int> 18, 16, 3, 39, 41, 6, 23, 45, 20, 18, 8, 2, 1, 0, 18, 3,…
## $ sex           <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Fe…
## $ salary        <int> 139750, 173200, 79750, 115000, 141500, 97000, 175000, 14…
```

The variable yrs.since.phd, yrs.service, and salary are numeric. The other three variables rank, discipline and sex are categorical variables.

3.

```
set.seed(1626)
train.index <- sample(c(1:nrow(Salaries)), nrow(Salaries)*0.6)
train.df <- Salaries[train.index, ]
valid.df <- Salaries[-train.index, ]
```

a. A model is built to make accurate predictions on unseen data. This data is the real-world data and partitioning allows us to simulate this scenario by keeping a portion of the data for validation.

If we use the entire dataset for an in-depth analysis before partitioning, we can leak information from the validation set into the training set as they are mixed together at this stage. We might accidentally use information from the validation set to make decisions about the model during training, which can lead to overly optimistic performance estimates. This can also lead to overfitting. Overfitting occurs, when a model learns the training data, including noise and peculiarities, too well. We also need to separate the dataset to evaluate the model's performance.

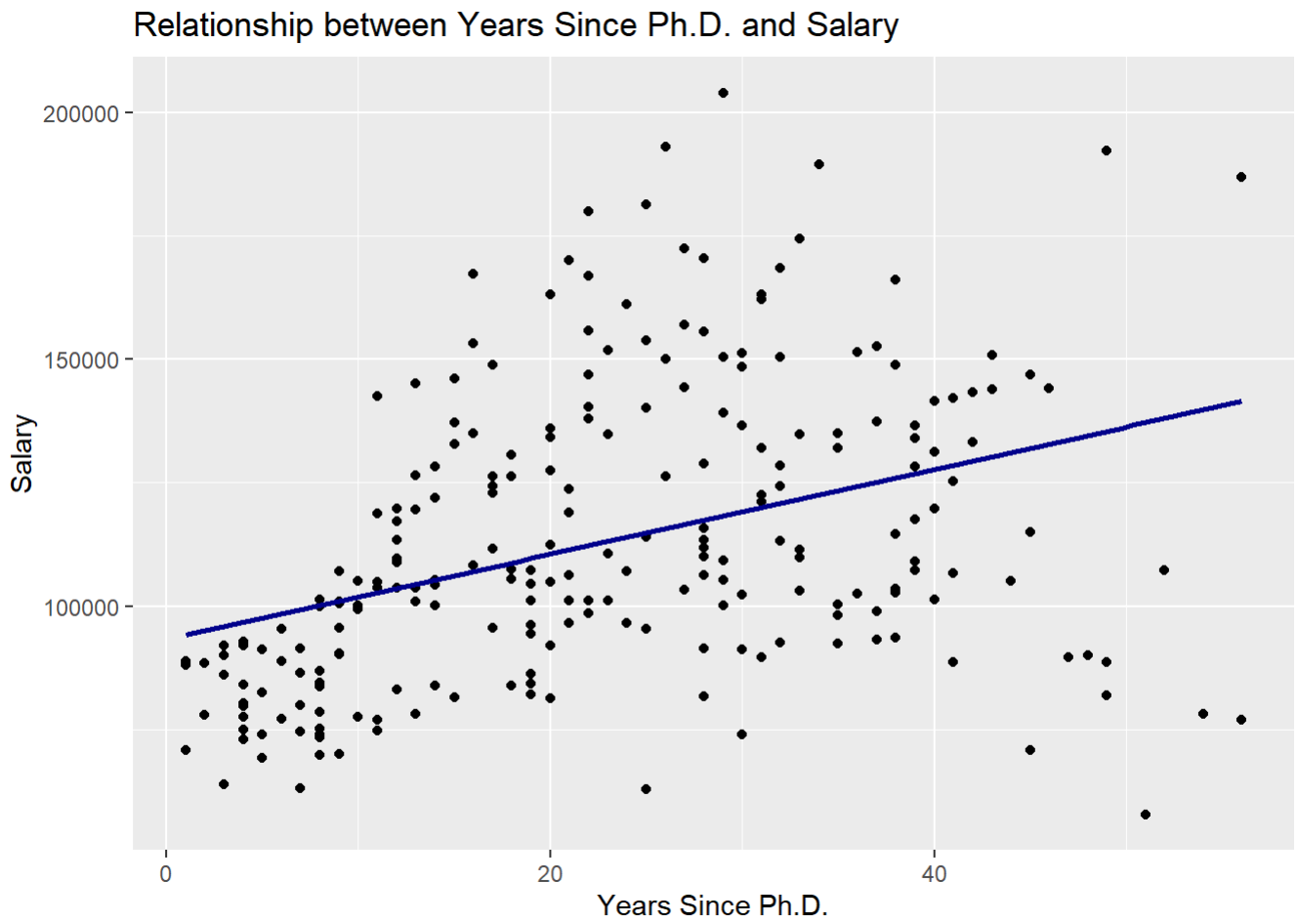
4.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
ggplot(data = train.df, aes(x = yrs.since.phd , y = salary)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue4") +
  labs(x = "Years Since Ph.D.", y = "Salary") +
  ggtitle("Relationship between Years Since Ph.D. and Salary")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The plot suggest that as the years since phd increases, the salary increases as well. The variables seem to have a positive correlation. The best fit line represents this notion. However, there are quite a few data points that do not follow this pattern. Quite a few of the points fall below the best fit line that have many years since phd but the salary is low. The highest salaries in the list do not have the highest number of years since phd. They seem to be in the middle.

I presumed a similar distribution but expected the highest values for the salary to fall more to the right side of the graph. This could be because there are less data on people who had many years since phd.

5.

```
correlation <- cor(train.df$yrs.since.phd, train.df$salary)
cor_test <- cor.test(train.df$yrs.since.phd, train.df$salary)
p_value <- cor_test$p.value
```

The results for these codes are- correlation : 0.37842439876464 p_value : 1.6077

The correlation between “Years Since PhD” and “Salary” is positive, indicating a modest linear relationship. It is not a strong correlation as it is not close to 1.

However, the correlation could not be statistically significant because of the high p-value. This indicates that there’s a chance the observed association could have happened by random chance.

6.

```
model <- lm(salary ~ yrs.since.phd, data = train.df)

summary(model)
```

```
##
## Call:
## lm(formula = salary ~ yrs.since.phd, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79354 -19447  -3923  16128  85767
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    93291.2     3594.0  25.958  < 2e-16 ***
## yrs.since.phd    860.1      136.9   6.281 1.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27560 on 236 degrees of freedom
## Multiple R-squared:  0.1432, Adjusted R-squared:  0.1396
## F-statistic: 39.45 on 1 and 236 DF,  p-value: 1.608e-09
```

Minimum residual: -79354 Maximum residual: 85767

a.

```
residuals <- resid(model)

highest_residual_index <- which.max(residuals)
observation_with_highest_residual <- train.df[highest_residual_index, ]

predicted_salary <- predict(model, observation_with_highest_residual)
predicted_salary
```

```
##      250
## 118232.7
```

```
print(observation_with_highest_residual)
```

```
##     rank discipline yrs.since.phd yrs.service  sex salary
## 250 Prof          A            29           7 Male 204000
```

The Prof's actual salary is 204000. The model predicted that it would be 118232.7. Residual = Actual Salary - Predicted Salary = 204,000 - 118,232.7 = $85,767.3

The residual for this observation is approximately 85,767.3. This means that the model's prediction was off by this amount for this specific observation. The positive sign indicates that the actual salary was higher than what the model predicted.

b.

```
lowest_residual_index <- which.min(residuals)
observation_with_lowest_residual <- train.df[lowest_residual_index, ]

predicted_salary_2 <- predict(model, observation_with_lowest_residual)
predicted_salary_2
```

```
##      283
## 137153.8
```

```
print(observation_with_lowest_residual)
```

```
##     rank discipline yrs.since.phd yrs.service  sex salary
## 283 Prof          A            51          51 Male  57800
```

The Prof's actual salary is 57800. The model predicted that it would be 137153.8. Residual = Actual Salary - Predicted Salary = 57800 - 137153.8 = -79,353.8

The residual for this observation is approximately -79,353.8. This means that the model's prediction was off by this amount for this specific observation. The negative sign indicates that the actual salary was lower than what the model predicted.

c. Because Years Since PhD alone does not accurately predict a professor's pay and does not account for all of the elements that affect a professor's compensation, the model is flawed. The model does not take into account additional factors that can have a substantial impact on compensation, such as area of specialty, record of publications, effectiveness as a teacher, and external funds.

Also, the relationship between Years Since PhD and salary may not be strictly linear. For instance, a professor's salary might increase more rapidly in the early years following their PhD, and then plateau or increase more slowly over time. The linear model might not capture this non-linear relationship.

9. From the Coefficients of our model, we can see that the intercept is 93291.2 and the slope is 860.1 or 860. If I use the standard equation of a straight line (y = mx + b), I can have the below equation as my model's regression equation. y= 93291.2 +860*x

Here x is Years Since PHD and y is Salary. Thus for any hypothetical year, we can find the Salary from this model.

```
x <- 5
y <- 93291.2 + 860 * x
y
```

```
## [1] 97591.2
```

If I take x as 5 years then y would be 97591.2.

10.

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
pred_model <- predict(model, train.df)
accuracy(pred_model, train.df$salary)
```

```
##                      ME     RMSE      MAE       MPE     MAPE
## Test set -3.069255e-11 27443.73 21549.72 -5.812143 19.90845
```

```
pred_model2 <- predict(model, valid.df)
accuracy(pred_model2, valid.df$salary)
```

```
##                ME     RMSE      MAE       MPE     MAPE
## Test set 3054.482 27680.08 21060.22 -3.012627 18.61182
```

```
mean(Salaries$salary)
```

```
## [1] 113706.5
```

We wish to avoid overfitting the model when using linear regression for prediction. There will be no practical purpose for the overfitted model. It is necessary to test the model against a validation dataset for this reason.

If we compare the ME (Mean Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MPE (Mean Percentage Error), and MAPE (Mean Absolute Percentage Error) of the model for both the training and validation data, we can see that the mean error for the validation set is quite a bit bigger than the training set. Besides that, all of the other numbers are not quite far off from each other.

The RMSE is 27443.73 for the training data and 27680.08 for the validation data., which means that, on average, the predictions deviate from the actual values by approximately 27443.73 for the training data and 27680.08 for the validation data. Thus the model is showing similar results.

The MAE is 21549.72 for the training data and 21060.22 for the validation data, which means that, on average, the absolute difference between the predicted and actual values is approximately 21549.72 for the training data and 21060.22 for the validation data.

The model is not very accurate as the RMSE and MAE values are quite large.

11.

```
standard_deviation <- sd(train.df$salary)
standard_deviation
```

```
## [1] 29711.11
```

The salary standard deviation in the training set is 29711.11, and the RMSE for the training data is 27443.73. Compared to the RMSE, the standard deviation is slightly larger. When the root mean square error (RMSE) is slightly above or equal to the standard deviation, it suggests that the model's predictions are not significantly better than utilizing the average income as a predictor.

If the RMSE is close to the standard deviation of salaries, it might indicate that there are important predictors missing from the model. This suggests that incorporating additional features or refining the existing model could lead to better predictions. If the RMSE is smaller than the standard deviation, it indicates that the model can provide meaningful predictions beyond just using the mean. This can be valuable for making informed decisions or understanding the factors that influence salaries.

● Multiple Linear Regression: 1.

```
str(train.df)
```

```
## 'data.frame':    238 obs. of  6 variables:
##  $ rank         : Factor w/ 3 levels "AsstProf","AssocProf",..: 2 3 2 3 3 3 3 3 3 3 ...
##  $ discipline   : Factor w/ 2 levels "A","B": 2 1 1 2 2 2 2 2 2 2 ...
##  $ yrs.since.phd: int  10 28 20 40 40 18 35 16 28 26 ...
##  $ yrs.service  : int  10 26 17 40 27 10 34 16 25 22 ...
##  $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 1 2 2 2 2 ...
##  $ salary       : int  99247 155500 81285 119700 101299 105450 92391 167284 111751 150000 ...
```

```
names(train.df)
```

```
## [1] "rank"          "discipline"    "yrs.since.phd" "yrs.service"
## [5] "sex"           "salary"
```

```
Salaries_numbers <- train.df[, c(3,4,6)]
cor(Salaries_numbers)
```

```
##               yrs.since.phd yrs.service    salary
## yrs.since.phd     1.0000000   0.9187046 0.3784244
## yrs.service       0.9187046   1.0000000 0.3280483
## salary            0.3784244   0.3280483 1.0000000
```

Among the six different variables, I am only using three as the other three are factors and are not very useful in making a correlation table. By looking at the correlations table, we can see that the variable Years Since PHD and Years in Service have a very high correlation of 0.9187. Multicollinearity might be an issue here. Thus I will be removing the Years Since PHD variable.

Multicollinearity occurs when there is a linear relationship between independent variables, which can make it difficult to separate their individual effects on the dependent variable.

2. Dummy variables are a set of binary (0 or 1) variables used in regression analysis to represent categorical data. They are often referred to as indicator variables or binary variables. Their function is to encode categorical variables so that they can be incorporated into a regression model.
3.

```
train.df <- train.df[,-3]
```

```
MLR_model <- lm(salary ~ rank + discipline + yrs.service + sex, data = train.df)

summary(MLR_model)
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + sex,
##     data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -58749 -13215  -1468  10436  82201
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    68796.3     6131.8  11.220  < 2e-16 ***
## rankAssocProf  15916.3     5104.0   3.118  0.00205 **
## rankProf       50820.6     4971.0  10.223  < 2e-16 ***
## disciplineB    13351.8     2942.2   4.538 9.12e-06 ***
## yrs.service     -119.3      136.6  -0.873  0.38330
## sexMale         3017.4     5258.4   0.574  0.56665
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22140 on 232 degrees of freedom
## Multiple R-squared:  0.4563, Adjusted R-squared:  0.4446
## F-statistic: 38.94 on 5 and 232 DF,  p-value: < 2.2e-16
```

```
lm.step <- step(MLR_model, direction = "backward")
```

```
## Start:  AIC=4768.43
## salary ~ rank + discipline + yrs.service + sex
##
##               Df  Sum of Sq        RSS    AIC
## - sex          1 1.6144e+08 1.1391e+11 4766.8
## - yrs.service  1 3.7410e+08 1.1412e+11 4767.2
## <none>                      1.1375e+11 4768.4
## - discipline   1 1.0097e+10 1.2385e+11 4786.7
## - rank         2 6.4874e+10 1.7862e+11 4871.8
##
## Step:  AIC=4766.77
## salary ~ rank + discipline + yrs.service
##
##               Df  Sum of Sq        RSS    AIC
## - yrs.service  1 3.3270e+08 1.1424e+11 4765.5
## <none>                      1.1391e+11 4766.8
## - discipline   1 9.9807e+09 1.2389e+11 4784.8
## - rank         2 6.5313e+10 1.7923e+11 4870.6
##
## Step:  AIC=4765.46
## salary ~ rank + discipline
##
##              Df  Sum of Sq        RSS    AIC
## <none>                     1.1424e+11 4765.5
## - discipline  1 1.0512e+10 1.2476e+11 4784.4
## - rank        2 9.1483e+10 2.0573e+11 4901.5
```

```
summary(lm.step)
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -61731 -13320  -1069  10801  84469
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      70961       4101  17.302  < 2e-16 ***
## rankAssocProf    14746       4871   3.027  0.00275 **
## rankProf         48570       4034  12.039  < 2e-16 ***
## disciplineB      13517       2913   4.640  5.8e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22100 on 234 degrees of freedom
## Multiple R-squared:  0.4539, Adjusted R-squared:  0.4469
## F-statistic: 64.84 on 3 and 234 DF,  p-value: < 2.2e-16
```

4.

a.

```
mean_salary <- mean(train.df$salary)
squared_diff <- (train.df$salary - mean_salary)^2
total_sum_of_squares <- sum(squared_diff)
total_sum_of_squares
```

```
## [1] 209211819738
```

b.

```
fitted_values <- predict(MLR_model)
mean_salary <- mean(train.df$salary)
squared_diff <- (fitted_values - mean_salary)^2
sum_of_squares_regression <- sum(squared_diff)
sum_of_squares_regression
```

```
## [1] 95461466045
```

c.

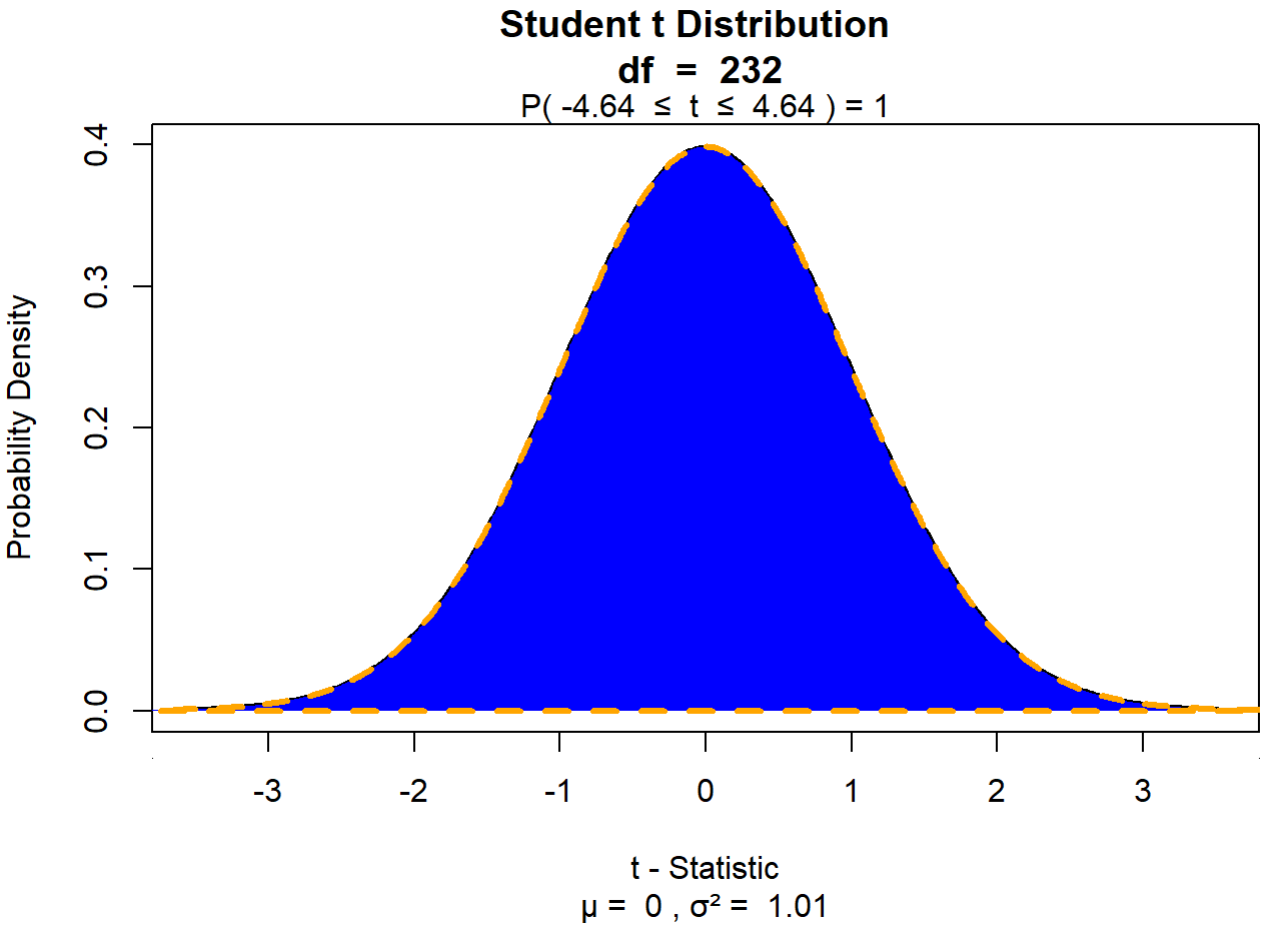
```
ssr_sst <- sum_of_squares_regression/total_sum_of_squares
ssr_sst
```

```
## [1] 0.456291
```

SSR/SST for the model is 0.456291. We can also see this number at the end of the summery for our model as Multiple R-squared: 0.4563.

5.

```
library(visualize)
visualize.t(stat=c(-4.64,4.64), df=232, section="bounded")
```



I choose discipline as the predictor the plot this graph. The amount of shaded region is 100%. Thus we cannot get a t value larger than the one we got.This imput variable meets the 95% threshold. This shaded area is the p-value associated with the t-value. In hypothesis testing, the p-value represents the probability of observing a t-value as extreme as the one calculated from the data, assuming the null hypothesis is true.

6.

```
summary_model <- summary(MLR_model)
f_statistic <- summary_model$fstatistic[1]
f_statistic
```

```
##    value
## 38.93976
```

In a multiple regression context, the F-statistic is calculated as the ratio of two mean square values. An F-statistic value of 38.93976 indicates that there is evidence to suggest that at least one of the independent variables in the model is contributing significantly to the explanatory power of the model.

7. I have assumed the following attributes for the fictional professor:

Rank: “Prof” Discipline: “A” Years of Service: 12 Sex: “Female”

```
new_data <- data.frame(rank = "Prof",
                       discipline = "A",
                       yrs.service = 12,
                       sex = "Female")

predicted_salary <- predict(MLR_model, newdata = new_data)

print(predicted_salary)
```

```
##        1
## 118185.1
```

8.

```
pred_model3 <- predict(MLR_model, train.df)
accuracy(pred_model3, train.df$salary)
```

```
##                      ME    RMSE      MAE       MPE     MAPE
## Test set -1.507776e-10 21861.9 16401.16 -3.217074 14.28403
```

```
pred_model4 <- predict(MLR_model, valid.df)
accuracy(pred_model4, valid.df$salary)
```

```
##                ME     RMSE      MAE       MPE     MAPE
## Test set 1022.718 23410.68 16709.48 -2.349496 14.19041
```

The model seems to perform fairly well on both the training and test sets. The ME values being close to zero suggest that, on average, the predictions are accurate.

The RMSE values (around 21861.9 and 23410.68) indicate the average magnitude of errors. They are not extremely high, suggesting that the model is reasonably accurate.

The MAPE values (around 14.28% and 14.19%) represent the average percentage error, indicating the model's relative accuracy.

Both the MLR and SLR models exhibit relatively low Mean Absolute Percentage Error (MAPE) values on both the training and test sets. This indicates that neither model is likely suffering from severe overfitting.

The MLR model generally outperforms the SLR model in terms of accuracy on both the training and test sets. This is evident from the lower RMSE, MAE, and MAPE values for the MLR model.

On the training set, the MLR model exhibits a slightly higher RMSE compared to the SLR model. This could be attributed to the added complexity of the MLR model, which may introduce some noise from the additional variables. However, the MLR model compensates for this by offering better performance on the test set, indicating better generalization to new data.