```
# load libraries
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
# load data
wine <- read.csv("C:/Users/mattm/Desktop/SchoolMyPC/SE 4375MachineLearning/Groupassnmt/winemag-data_first150k.csv", string
sAsFactors = FALSE)

# subset data
wine_sub <- wine[1:15000, ]

# remove rows with missing values in the price column
wine_sub <- wine_sub[!is.na(wine_sub$price),]
wine_sub <- wine_sub[, -1]
wine_sub <- wine_sub[!is.na(wine_sub$region_1),]
wine_sub <- wine_sub[!is.na(wine_sub$region_2),]

# create train and test sets
set.seed(123)
trainIndex <- createDataPartition(wine_sub$price, p = 0.7, list = FALSE, times = 1)
train <- wine_sub[trainIndex, ]
test <- wine_sub[-trainIndex, ]

summary(train)
```

```
##    country          description        designation            points
##  Length:9850        Length:9850        Length:9850        Min.   : 80.00
##  Class :character   Class :character   Class :character   1st Qu.: 87.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 89.00
##                                                           Mean   : 88.75
##                                                           3rd Qu.: 91.00
##                                                           Max.   :100.00
##      price           province           region_1           region_2
##  Min.   :   4.00   Length:9850        Length:9850        Length:9850
##  1st Qu.:  18.00   Class :character   Class :character   Class :character
##  Median :  27.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  36.51
##  3rd Qu.:  45.00
##  Max.   :2013.00
##    variety             winery
##  Length:9850        Length:9850
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
summary(test)
```

```
##    country           description        designation            points     
##  Length:4218        Length:4218        Length:4218        Min.   :80.00  
##  Class :character   Class :character   Class :character   1st Qu.:87.00  
##  Mode  :character   Mode  :character   Mode  :character   Median :88.00  
##                                                           Mean   :88.67  
##                                                           3rd Qu.:91.00  
##                                                           Max.   :99.00  
##      price           province           region_1           region_2        
##  Min.   :   6.00   Length:4218        Length:4218        Length:4218       
##  1st Qu.:  18.00   Class :character   Class :character   Class :character  
##  Median :  26.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :  37.02                                                           
##  3rd Qu.:  45.00                                                           
##  Max.   :1100.00                                                           
##    variety             winery         
##  Length:4218        Length:4218       
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
```

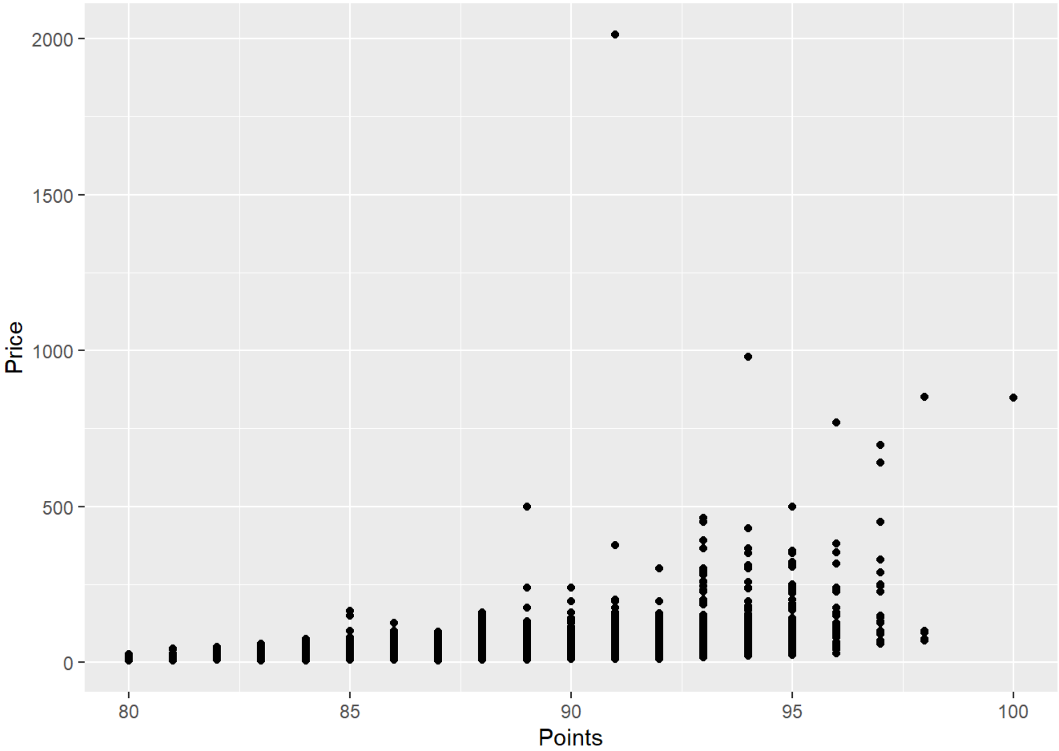
```
dim(train)
```

```
## [1] 9850   10
```

```
dim(test)
```

```
## [1] 4218   10
```

```
ggplot(train, aes(x = points, y = price)) +
  geom_point() +
  labs(x = "Points", y = "Price")
```



Here we look at a ggplot graph showing us the relationship of price to points, the score given to the wine.

While score does trend up with price there are times when overpaying did not result in a better wine overall.

```
# create linear regression model
lm_fit <- train(price ~ points, data = train, method = "lm")

# create kNN regression model
knn_fit <- knnreg(points ~ ., data = train, method = "knn", trControl = trainControl(method = "cv", number = 10), tuneGrid = expand.grid(k = 1:30))

# create decision tree regression model
dt_fit <- train(price ~ points, data = train, method = "rpart", trControl = trainControl(method = "cv", number = 10), tuneGrid = expand.grid(cp = seq(0, 0.1, by = 0.01)))


# make predictions
lm_pred <- predict(lm_fit, newdata = test)
# select columns to include in test data
test_cols <- c("country", "description", "designation", "price", "points","region_1","region_2", "province", "variety", "winery")


# subset test data to selected columns
test_subset <- test[, test_cols]

# make predictions using kNN model on subset of test data
#knn_pred <- predict(knn_fit, newdata = test_subset)

dt_pred <- predict(dt_fit, newdata = test)

# calculate performance metrics
lm_rmse <- RMSE(lm_pred, test$price)
lm_rsq <- cor(lm_pred, test$price)^2
lm_mae <- MAE(lm_pred, test$price)

#knn_rmse <- RMSE(knn_pred, test$price)
#knn_rsq <- cor(knn_pred, test$price)^2
#knn_mae <- MAE(knn_pred, test$price)

dt_rmse <- RMSE(dt_pred, test$price)
dt_rsq <- cor(dt_pred, test$price)^2
dt_mae <- MAE(dt_pred, test$price)

# print results
cat("Linear Regression:\n")
```

```
## Linear Regression:
```

```
cat("RMSE = ", lm_rmse, "\n")
```

```
## RMSE =  39.74468
```

```
cat("R-squared = ", lm_rsq, "\n")
```

```
## R-squared =  0.1600689
```

```
cat("MAE = ", lm_mae, "\n\n")
```

```
## MAE =  18.08295
```

```
#cat("kNN Regression:\n")
#cat("RMSE = ", knn_rmse, "\n")
#cat("R-squared = ", knn_rsq, "\n")
#cat("MAE = ", knn_mae, "\n\n")

cat("Decision Tree Regression:\n")
```

```
## Decision Tree Regression:
```

```
cat("RMSE = ", dt_rmse, "\n")
```

```
## RMSE =  38.02449
```

```
cat("R-squared = ", dt_rsq, "\n")
```

```
## R-squared =  0.2328322
```

```
cat("MAE = ", dt_mae, "\n")
```

```
## MAE =  16.70968
```

Comparing the results of the linear regression and decision tree regression models, we see that the decision tree regression model performs slightly better than the linear regression model in terms of RMSE and R-squared, but the linear regression model performs slightly better in terms of MAE.

This is likely due to the differences in the algorithms used by each model. Linear regression models assume a linear relationship between the independent and dependent variables, and try to fit a straight line to the data. Decision tree regression models, on the other hand, recursively split the data into smaller groups based on the independent variables, and fit a decision tree to the data.

In this case, it’s possible that the relationship between the independent and dependent variables is not entirely linear, but can be better captured by the decision tree regression model. However, the decision tree model may also be more prone to overfitting the data, as it can easily become too complex and fit the noise in the data rather than the underlying patterns.

Overall, both models have their strengths and weaknesses, and the choice of which to use may depend on the specific characteristics of the data and the goals of the analysis.