Assignment Two

Ben Waetford

2022-08

edX HarvardX PH125.9x

Capstone Project: IDV Learners

Introduction

Purpose

The HardardX Data Science Professional Certificate requires students to complete nine courses. This paper (the Individual Assignment) is intended to satisfy the final requirement of the ninth course. The Individual Assignment requires students to apply machine learning techniques (beyond simple linear regression) to answer a question of their own choice using data that is *not* well known. Thus, for this Individual Assignment, I have chosen to use a data set from a hypothetical record label known as Acme Musical Sounds Inc.

Data

Data Selection The data set employed in this Individual Assignment represents data from Acme Musical Sounds Inc.'s Enterprise Resource System.

Data Definitions Acme Inc. regularly enters into contracts (jobs) to produce soundtracks for various clients. Each contract is segmented by a number of factors. The project factors are independent variables. In addition to the independent variables, Acme Musical Sounds Inc. supplied one dependent variable, "Result" which describes whether Acme Musical Sounds Inc. earned a lot of money (Big win), earned some money (win), broke even (close enough) lost money (loss), or lost a lot of money (big loss) as a result of completing the agreed upon musical production job.

The data set:

Variable	Description
Client Base	Each project is associated with one of three different client types.
Contract Type	One of five different contract delivery frameworks employed during contract negotiation.
State	The geographic region that the customer requires the project be delivered within.
Joint Venture	An indicator that displays 'yes' if Acme is the sole owner of the project.

Variable	Description
Recurring Revenue	An indicator that displays 'Y' if the project is renewed on a regular basis without the need for the client to go to market and seek competitive bids from other prospective vendors or suppliers.
Delivery Model	One of six different models used to manage the delivery of the project.
Sector	The group within Acme Musical Sounds Inc. that leads the service delivery on the project.
Service Focus	One of 24 service specialty groups within Acme Musical Sounds Inc. that manages the customer relationship.
Duration	A number that represents the duration of the project. A higher number represents a longer duration.
Year	The year that the project began.
OM	A number from one to 10 that represents the relative size of the job, one being the smallest that Acme Musical Sounds Inc. would perform, 10 being the largest.
Result	One of five outcomes: big loss, loss, close enough, win, big win.
Category	An engineered feature that groups the five 'Result' variables into two groups—'Better than expected' (those jobs with an actual profit that was greater than expected) and 'Worse than expected' (those jobs that lost money).

Research Question

The owner of Acme Musical Sounds Inc. continues to pursue opportunities to grow their business by delivering new projects for new customers. The company wishes to know if out-of-the-box (untuned) Machine Learning models might successfully predict jobs that will result in a profit margin that is better than expected. A successful prediction, for the purpose of this assignment, is any prediction with an accuracy score greater than the no information rate (0.720).

Research Activities

Source data was imported, and checked for NAs. The data was then transformed into factors, a dependent feature was engineered. Six classification models were fitted and the accuracy of each was reviewed to determine if any accuracy scores were greater than the no information rate.

Results Summary

Two models performed as well as the no information rate. Three models out-performed the no information rate by a small margin (by 0.01 and 0.02 overall accuracy). One model performed worse than the no information rate. Thus, the null hypothesis, that Acme Musical Sounds Inc.'s data sample has little to no predictive potential, has been accepted.

Future analysis should be performed on a larger data set, should incorporate parameter tuning, apply ensemble methods, and explore options to control prevalence. Such additional actions may improve upon the accuracy scores reported in this assignment.

Method and Analysis

Libraries

```
#load libraries
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl", repos = "http://cran.us.r-project.org")
if(!require(rpart.plot)) install.packages("rpart.plot", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(naivebayes)) install.packages("naivebayes", repos = "http://cran.us.r-project.org")
if(!require(partykit)) install.packages("partykit", repos = "http://cran.us.r-project.org")
```

Data Cleaning and Preparation

Most data were categorical, but coded as continuous (numeric). All data was transformed into factors such that they would be understood by R as being categorical data–suitable for analysis by classification trees.

Train and test sets were created. 70% of data were allocated for training and the remaining 30% were used for testing. 70% split was chosen since the data set is small and a larger training set would have resulted in a test set that did not contain all factors/levels present in the training set.

```
#read and format data
my_data <- read_excel("data.xlsx")
head(my_data)
```

```
## # A tibble: 6 x 12
##   Result  External_mgmt Client_base Contract_type Contractor_type Delivery_model
##   <chr>   <chr>         <chr>               <dbl>           <dbl>          <dbl>
## 1 Loss    no            Y                       1               1              1
## 2 Loss    no            Y                       2               1              2
## 3 Close ~ no            Y                       1               1              1
## 4 Close ~ no            Y                       1               1              3
## 5 Win     no            Y                       2               1              1
## 6 Win     no            Y                       2               1              1
## # ... with 6 more variables: State <dbl>, Sector <chr>, Recur_revenue <chr>,
## #   Service_focus <dbl>, start_year <dbl>, OM <dbl>
```

```
dat <- my_data %>%
  mutate(result = as.factor(Result),
         ext_mgmt = as.factor(External_mgmt),
         client_b = as.factor(Client_base),
         contract_t = as.factor(Client_base),
         contractor_t = as.factor(Contractor_type),
         delivery_m = as.factor(Delivery_model),
         state = as.factor(State),
         sector = as.factor(Sector),
         recur_r = as.factor(Recur_revenue),
         service_f = as.factor(Service_focus),
         syear = as.factor(start_year),
         order_m = as.factor(OM),
         Result = NULL,
         External_mgmt= NULL,
         Client_base = NULL,
         Contract_type = NULL,
         Contractor_type = NULL,
         Delivery_model= NULL,
         State = NULL,
         Sector = NULL,
         Recur_revenue = NULL,
         Service_focus = NULL,
         start_year = NULL,
         OM = NULL)
```

Key concept: in this assignment, the term [big] winner refers to jobs that earned more profit than planned. [Big] loser refers to jobs that earned less profit than planned. Close enough refers to jobs that are neither winners or losers.

The 'result' factor has five levels, each of increasing financial impact, from big loser to big winner. The levels were reordered so that when result is plotted on an x-axis, they can be easily read from left-to-right by order of financial impact (negative to positive).

```
dat$result <- factor(dat$result, levels = c('Big Loss', 'Loss', 'Close Enough', 'Win', 'Big Win'))
```

A new variable ('category') was generated based on the 'results' variable. 'Category' is the dependent variable in Machine Learning models used in this assignment.

```
dat <- dat %>%
  mutate(category = as.factor(ifelse(dat$result %in% c('Close Enough', 'Win', 'Big Win'), "Better than 
  )
```

Finally, the data set is split into a test and training set. And data frame was created to store the overall accuracy score of each Machine Learning model.

```
# generate training and test sets
set.seed(1976, sample.kind = 'Rounding') #make repeatable
ind <- sample(2, nrow(dat), replace = T, prob = c(0.7, 0.3)) #generate indexes for each set
train <- dat[ind==1,] #create training set
test <- dat[ind==2,] #create test set

#create a table that will be used to record the accuracy of each model, the rate to beat is the no info
accuracy_table <- data_frame(Model = "No information rate", Rate = 0.7199)
```

Data Exploration and Insights

The data contains no NAs, and contains 1031 rows.

```
#check for na
na.s <- sapply(dat, {function(x) any(is.na(x))})
knitr::kable(na.s)
```

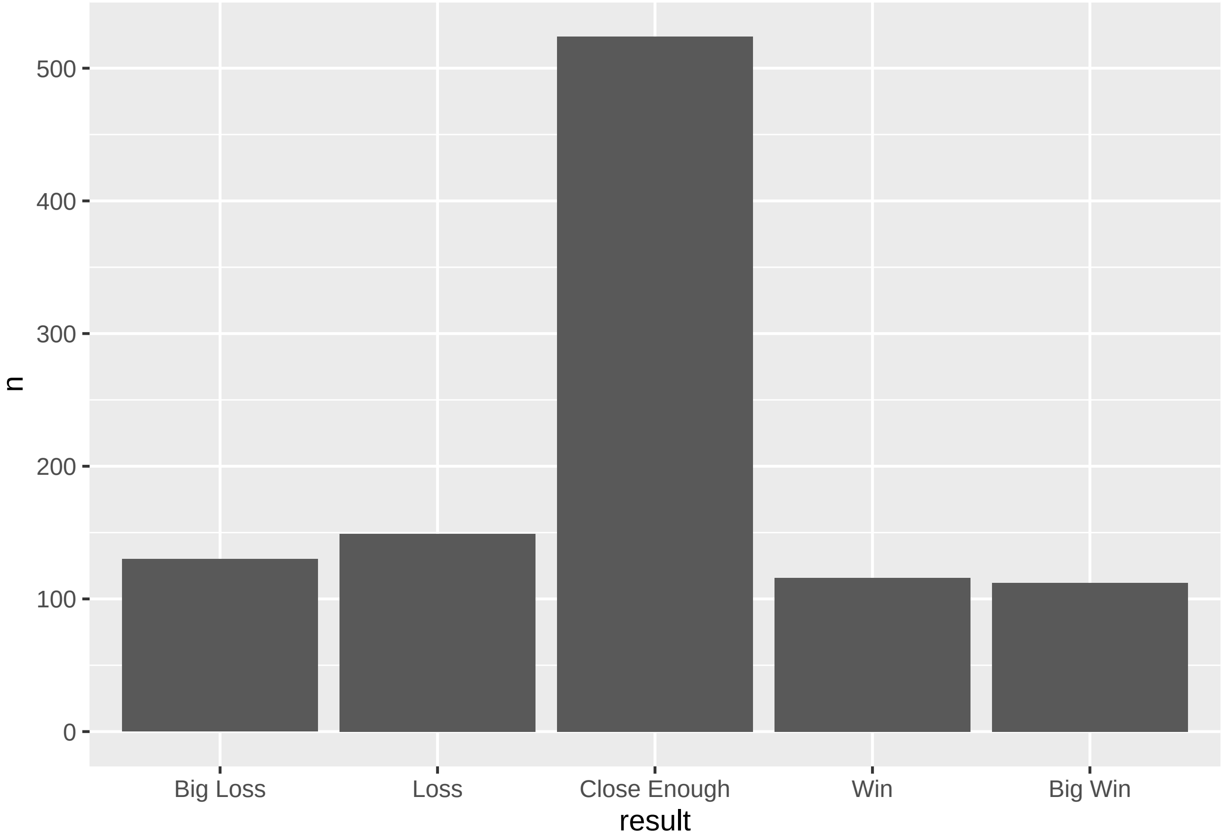
	x
result	FALSE
ext_mgmt	FALSE
client_b	FALSE
contract_t	FALSE
contractor_t	FALSE
delivery_m	FALSE
state	FALSE
sector	FALSE
recur_r	FALSE
service_f	FALSE
syear	FALSE
order_m	FALSE
category	FALSE

```
#How many rows of data?
nrow(dat)
```

```
## [1] 1031
```

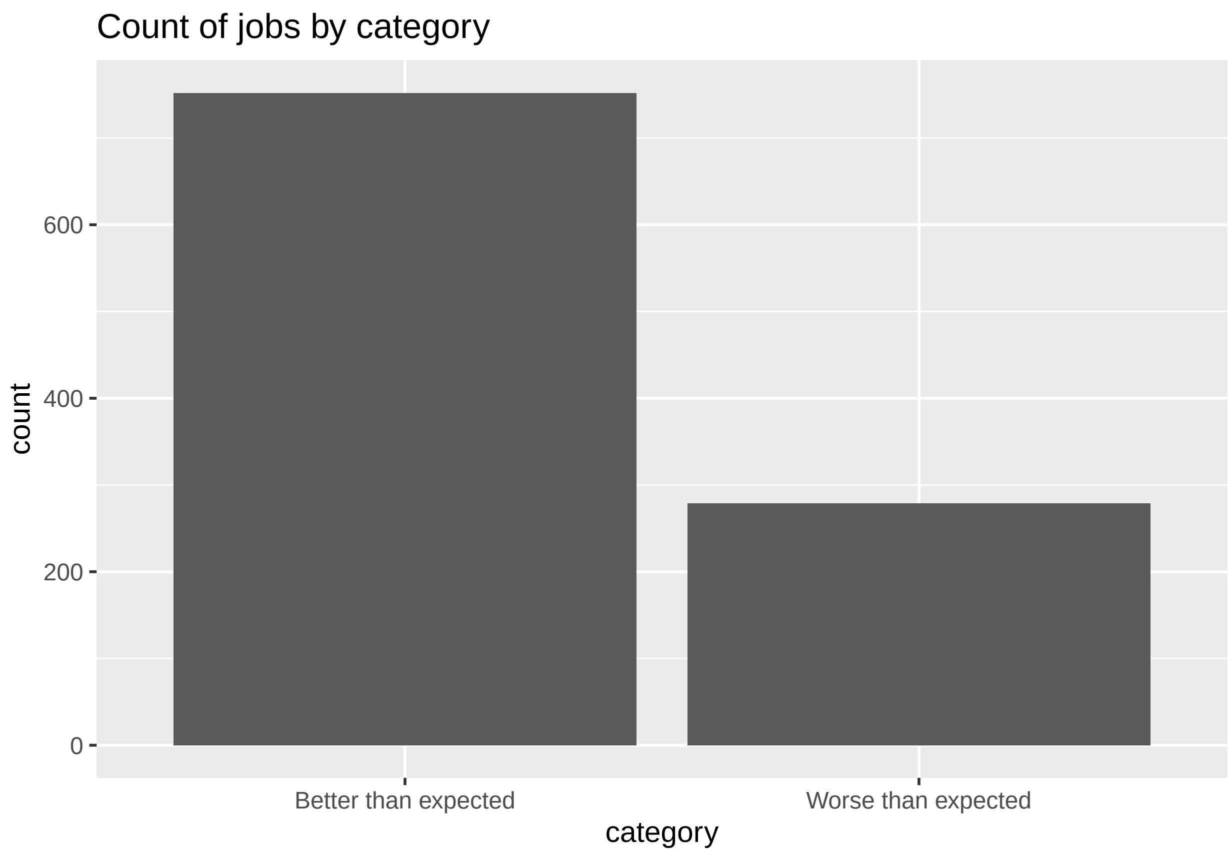
Most jobs are neither lose or win money, instead, more than 500 jobs ‘s financial results are ’close enough’ .

```
dat %>%
  group_by(result) %>%
  summarise(n = n()) %>%
  ggplot(aes(result, n)) +
  geom_col()
```

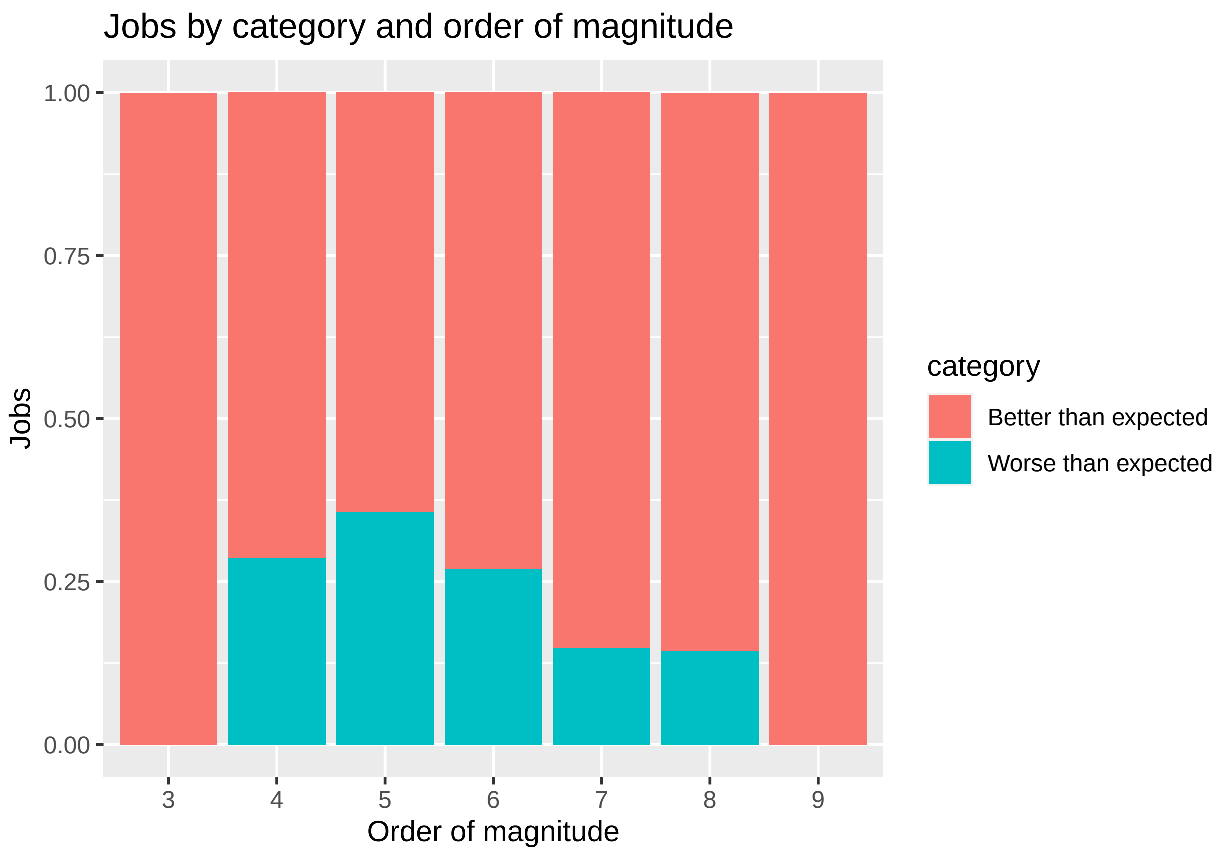


When summarized further–by ‘category’, we notice that approximately one-third of jobs perform worse than expected, and that jobs with low order-of-magnitude financial impact are slightly more likely to be losers than bigger jobs.

```
ggplot(dat, aes(category)) +
  geom_bar() +
  ggtitle("Count of jobs by category")
```



```
ggplot(dat, aes(order_m)) +
  geom_bar(position = "fill", aes(fill = category)) +
  ggtitle("Jobs by category and order of magnitude") +
  xlab("Order of magnitude") +
  ylab("Jobs")
```



Models

Before running the Machine Learning models, the variable ‘result’ must be removed from the test and train data sets becuase the engineered feature ‘category’ is *dependent* on the “result’ variable.

```
train <- train %>% select(-c(result))
test <- test %>% select(-c(result))
```

Six models were trained using the train data set. Once trained, the predict function was used to predict the dependent variable in the test set. Confusion matrix was used to compare the predicted and actual values in the test set, and the overall accuracy scores were written to a results table.

```
#randomForest's randomForest function
set.seed(1979, sample.kind = 'Rounding') #make results repeatable
rf <- randomForest(category ~ ., data = train) #fit model
p.rf <-predict(rf, test) #prediction
rF.a <- confusionMatrix(p.rf, test$category)$overall["Accuracy"] #compute accuracy
accuracy_table <- bind_rows(accuracy_table, data_frame(Model="randomForest", Rate  = rF.a)) #add score


#caret's train function
set.seed(1979, sample.kind = 'Rounding') #make results repeatable
caret_train <- train(category ~., data = train, method = "rf") #fit model
p.caret <-predict(caret_train, test) #prediction
cm.caretrf <- confusionMatrix(p.caret, test$category)$overall["Accuracy"] #compute accuracy
```

```
accuracy_table <- bind_rows(accuracy_table, data_frame(Model="caret rf", Rate  = cm.caretrf)) #add scor

####rpart random forest
set.seed(1976, sample.kind = 'Rounding') #make results repeatable
rf_rpart <- rpart(category ~ ., train) #fit model
p.rpart <- predict(rf_rpart, test, type = 'class') #prediction
cm.rpart <- confusionMatrix(p.caret, test$category)$overall["Accuracy"] #compute accuracy
accuracy_table <- bind_rows(accuracy_table, data_frame(Model="rpart", Rate  = cm.rpart)) #add score to

####ctree
set.seed(1976, sample.kind = 'Rounding') #make results repeatable
t.ctree <- ctree(category ~., data = train) #fit model
p.ctree <-predict(t.ctree, test, type = 'response') #prediction
cm.ctree <- confusionMatrix(p.ctree, test$category)$overall["Accuracy"] #compute accuracy
accuracy_table  <- bind_rows(accuracy_table, data_frame(Model="ctree", Rate  = cm.ctree)) #add score to


#knn
set.seed(1976, sample.kind = 'Rounding') #make results repeatable
c.knn <- train_knn <- train(category ~ ., method = "knn", train) #fit model
p.knn <- predict(c.knn, test) #prediction
cm.knn <- confusionMatrix(p.knn, test$category)$overall["Accuracy"] #compute accuracy
accuracy_table  <- bind_rows(accuracy_table, data_frame(Model="knn", Rate  = cm.knn)) #add score to acc

##Naive bayes
set.seed(1976, sample.kind = 'Rounding') #make results repeatable
nb <- naive_bayes(category ~ ., data = train, laplace = 1) #fit model
p.nb <- predict(nb, test, type = 'class') #prediction
cm.nb <- confusionMatrix(p.nb, test$category)$overall["Accuracy"] #compute accuracy
accuracy_table  <- bind_rows(accuracy_table,data_frame(Model="Naive Bayes",Rate  = cm.nb)) #add score t
```

Results and Performance

The result table:

```
accuracy_table
```

```
## # A tibble: 7 x 2
##   Model                Rate
##   <chr>               <dbl>
## 1 No information rate 0.720
## 2 randomForest        0.736
## 3 caret rf            0.720
## 4 rpart               0.720
## 5 ctree               0.730
## 6 knn                 0.736
## 7 Naive Bayes         0.707
```

```
#accuracy table summary
summary(accuracy_table)
```

```
##     Model                Rate
##  Length:7           Min.   :0.7068
##  Class :character   1st Qu.:0.7199
##  Mode  :character   Median :0.7199
##                     Mean   :0.7241
##                     3rd Qu.:0.7329
##                     Max.   :0.7362
```

The out-of-the-box randomForest and knn have the best overall accuracy scores–but those scores are only marginally better than the no information rate.

However, when the sensitivity and specificity of the two best performing models are reviewed, we see that the specificity rates of each is low, meaning that even if we did wish to use either of these models (for a very minimal improvement over the no information rate), we would be presented with a large number of false positives.

```
#most accurate models sensitivity and specificity
#randomForest
confusionMatrix(p.rf, test$category)$byClass[1:2]
```

```
## Sensitivity Specificity
##   0.9230769   0.2558140
```

```
#knn
confusionMatrix(p.knn, test$category)$byClass[1:2]
```

```
## Sensitivity Specificity
##   0.9457014   0.1976744
```

The best out-of-the-box models examined in this Individual Assignment are only marginally better than the no information rate, and provide little value over guessing which jobs will perform well.

Conclusion

Summary

Six out-of-the-box Machine Learning models were used to process a small data set provided by Acme Musical Sounds Inc. The purpose of this assignment was to determine if that data set could be used to train Machine Learning models that, out-of-the-box, would perform better than the no information rate.

The data set was very small (<2000 rows of data), and without any parameter tuning, and without running any ensemble methods, Machine Learning was found to be of no assistance to Acme Musical Sounds Inc.'s management team.

Impact

This assignment found that Machine Learning models will only provide benefit if they are trained with meaningful data. This finding implies that the results of Machine Learning won't always be a 'magic hammer' that can solve any problem thrown at it.

Limitations

This assignment was performed for the sole purpose of meeting requirements of course 125.9. Nothing should be inferred from the data set nor the conclusions, since Acme Musical Sounds Inc. is a fictional company.

Future Considerations

An increased data sample size, coupled with exercises that tune the Machine Learning model parameters, and combining model results using an ensemble methodology will likely provide results that are more reliable that those created for this assignment.