Project (Naive bayes)

Helee Rana

4/29/2020

require(tidyverse)

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.2.1 ✓ purrr 0.3.3  
## ✓ tibble 2.1.3 ✓ dplyr 0.8.4  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

activity = read\_csv("/Users/heleerana/Desktop/Kaggle Red Hat/predicting-red-hat-business-value/act\_train.csv")

## Parsed with column specification:  
## cols(  
## people\_id = col\_character(),  
## activity\_id = col\_character(),  
## date = col\_date(format = ""),  
## activity\_category = col\_character(),  
## char\_1 = col\_character(),  
## char\_2 = col\_character(),  
## char\_3 = col\_character(),  
## char\_4 = col\_character(),  
## char\_5 = col\_character(),  
## char\_6 = col\_character(),  
## char\_7 = col\_character(),  
## char\_8 = col\_character(),  
## char\_9 = col\_character(),  
## char\_10 = col\_character(),  
## outcome = col\_double()  
## )

people = read\_csv("/Users/heleerana/Desktop/Kaggle Red Hat/predicting-red-hat-business-value/people.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_logical(),  
## people\_id = col\_character(),  
## char\_1 = col\_character(),  
## group\_1 = col\_character(),  
## char\_2 = col\_character(),  
## date = col\_date(format = ""),  
## char\_3 = col\_character(),  
## char\_4 = col\_character(),  
## char\_5 = col\_character(),  
## char\_6 = col\_character(),  
## char\_7 = col\_character(),  
## char\_8 = col\_character(),  
## char\_9 = col\_character(),  
## char\_38 = col\_double()  
## )

## See spec(...) for full column specifications.

test=read\_csv("/Users/heleerana/Desktop/Kaggle Red Hat/predicting-red-hat-business-value/act\_test.csv")

## Parsed with column specification:  
## cols(  
## people\_id = col\_character(),  
## activity\_id = col\_character(),  
## date = col\_date(format = ""),  
## activity\_category = col\_character(),  
## char\_1 = col\_character(),  
## char\_2 = col\_character(),  
## char\_3 = col\_character(),  
## char\_4 = col\_character(),  
## char\_5 = col\_character(),  
## char\_6 = col\_character(),  
## char\_7 = col\_character(),  
## char\_8 = col\_character(),  
## char\_9 = col\_character(),  
## char\_10 = col\_character()  
## )

## 

## Joining people and activity dataset

maindata = activity %>% left\_join(people , by =c('people\_id'='people\_id'))  
maindata%>% head(10)

## # A tibble: 10 x 55  
## people\_id activity\_id date.x activity\_catego… char\_1.x char\_2.x char\_3.x  
## <chr> <chr> <date> <chr> <chr> <chr> <chr>   
## 1 ppl\_100 act2\_17349… 2023-08-26 type 4 <NA> <NA> <NA>   
## 2 ppl\_100 act2\_24340… 2022-09-27 type 2 <NA> <NA> <NA>   
## 3 ppl\_100 act2\_34040… 2022-09-27 type 2 <NA> <NA> <NA>   
## 4 ppl\_100 act2\_36512… 2023-08-04 type 2 <NA> <NA> <NA>   
## 5 ppl\_100 act2\_41090… 2023-08-26 type 2 <NA> <NA> <NA>   
## 6 ppl\_100 act2\_898576 2023-08-04 type 4 <NA> <NA> <NA>   
## 7 ppl\_1000… act2\_12334… 2022-11-23 type 2 <NA> <NA> <NA>   
## 8 ppl\_1000… act2\_16234… 2022-11-23 type 2 <NA> <NA> <NA>   
## 9 ppl\_1000… act2\_11115… 2023-02-07 type 2 <NA> <NA> <NA>   
## 10 ppl\_1000… act2\_11774… 2023-06-28 type 2 <NA> <NA> <NA>   
## # … with 48 more variables: char\_4.x <chr>, char\_5.x <chr>, char\_6.x <chr>,  
## # char\_7.x <chr>, char\_8.x <chr>, char\_9.x <chr>, char\_10.x <chr>,  
## # outcome <dbl>, char\_1.y <chr>, group\_1 <chr>, char\_2.y <chr>,  
## # date.y <date>, char\_3.y <chr>, char\_4.y <chr>, char\_5.y <chr>,  
## # char\_6.y <chr>, char\_7.y <chr>, char\_8.y <chr>, char\_9.y <chr>,  
## # char\_10.y <lgl>, char\_11 <lgl>, char\_12 <lgl>, char\_13 <lgl>,  
## # char\_14 <lgl>, char\_15 <lgl>, char\_16 <lgl>, char\_17 <lgl>, char\_18 <lgl>,  
## # char\_19 <lgl>, char\_20 <lgl>, char\_21 <lgl>, char\_22 <lgl>, char\_23 <lgl>,  
## # char\_24 <lgl>, char\_25 <lgl>, char\_26 <lgl>, char\_27 <lgl>, char\_28 <lgl>,  
## # char\_29 <lgl>, char\_30 <lgl>, char\_31 <lgl>, char\_32 <lgl>, char\_33 <lgl>,  
## # char\_34 <lgl>, char\_35 <lgl>, char\_36 <lgl>, char\_37 <lgl>, char\_38 <dbl>

## Data Preprocessing

## Replacing NA values with -1

maindata1 = maindata %>% replace(is.na(.), -1)  
maindata1 %>% head(10)

## # A tibble: 10 x 55  
## people\_id activity\_id date.x activity\_catego… char\_1.x char\_2.x char\_3.x  
## <chr> <chr> <date> <chr> <chr> <chr> <chr>   
## 1 ppl\_100 act2\_17349… 2023-08-26 type 4 -1 -1 -1   
## 2 ppl\_100 act2\_24340… 2022-09-27 type 2 -1 -1 -1   
## 3 ppl\_100 act2\_34040… 2022-09-27 type 2 -1 -1 -1   
## 4 ppl\_100 act2\_36512… 2023-08-04 type 2 -1 -1 -1   
## 5 ppl\_100 act2\_41090… 2023-08-26 type 2 -1 -1 -1   
## 6 ppl\_100 act2\_898576 2023-08-04 type 4 -1 -1 -1   
## 7 ppl\_1000… act2\_12334… 2022-11-23 type 2 -1 -1 -1   
## 8 ppl\_1000… act2\_16234… 2022-11-23 type 2 -1 -1 -1   
## 9 ppl\_1000… act2\_11115… 2023-02-07 type 2 -1 -1 -1   
## 10 ppl\_1000… act2\_11774… 2023-06-28 type 2 -1 -1 -1   
## # … with 48 more variables: char\_4.x <chr>, char\_5.x <chr>, char\_6.x <chr>,  
## # char\_7.x <chr>, char\_8.x <chr>, char\_9.x <chr>, char\_10.x <chr>,  
## # outcome <dbl>, char\_1.y <chr>, group\_1 <chr>, char\_2.y <chr>,  
## # date.y <date>, char\_3.y <chr>, char\_4.y <chr>, char\_5.y <chr>,  
## # char\_6.y <chr>, char\_7.y <chr>, char\_8.y <chr>, char\_9.y <chr>,  
## # char\_10.y <lgl>, char\_11 <lgl>, char\_12 <lgl>, char\_13 <lgl>,  
## # char\_14 <lgl>, char\_15 <lgl>, char\_16 <lgl>, char\_17 <lgl>, char\_18 <lgl>,  
## # char\_19 <lgl>, char\_20 <lgl>, char\_21 <lgl>, char\_22 <lgl>, char\_23 <lgl>,  
## # char\_24 <lgl>, char\_25 <lgl>, char\_26 <lgl>, char\_27 <lgl>, char\_28 <lgl>,  
## # char\_29 <lgl>, char\_30 <lgl>, char\_31 <lgl>, char\_32 <lgl>, char\_33 <lgl>,  
## # char\_34 <lgl>, char\_35 <lgl>, char\_36 <lgl>, char\_37 <lgl>, char\_38 <dbl>

## 

## Converting Boolean values to numeric that is 0s and 1s

maindata2 = maindata1 %>% mutate\_if(is.logical,as.numeric)  
maindata2%>%head(10)

## # A tibble: 10 x 55  
## people\_id activity\_id date.x activity\_catego… char\_1.x char\_2.x char\_3.x  
## <chr> <chr> <date> <chr> <chr> <chr> <chr>   
## 1 ppl\_100 act2\_17349… 2023-08-26 type 4 -1 -1 -1   
## 2 ppl\_100 act2\_24340… 2022-09-27 type 2 -1 -1 -1   
## 3 ppl\_100 act2\_34040… 2022-09-27 type 2 -1 -1 -1   
## 4 ppl\_100 act2\_36512… 2023-08-04 type 2 -1 -1 -1   
## 5 ppl\_100 act2\_41090… 2023-08-26 type 2 -1 -1 -1   
## 6 ppl\_100 act2\_898576 2023-08-04 type 4 -1 -1 -1   
## 7 ppl\_1000… act2\_12334… 2022-11-23 type 2 -1 -1 -1   
## 8 ppl\_1000… act2\_16234… 2022-11-23 type 2 -1 -1 -1   
## 9 ppl\_1000… act2\_11115… 2023-02-07 type 2 -1 -1 -1   
## 10 ppl\_1000… act2\_11774… 2023-06-28 type 2 -1 -1 -1   
## # … with 48 more variables: char\_4.x <chr>, char\_5.x <chr>, char\_6.x <chr>,  
## # char\_7.x <chr>, char\_8.x <chr>, char\_9.x <chr>, char\_10.x <chr>,  
## # outcome <dbl>, char\_1.y <chr>, group\_1 <chr>, char\_2.y <chr>,  
## # date.y <date>, char\_3.y <chr>, char\_4.y <chr>, char\_5.y <chr>,  
## # char\_6.y <chr>, char\_7.y <chr>, char\_8.y <chr>, char\_9.y <chr>,  
## # char\_10.y <dbl>, char\_11 <dbl>, char\_12 <dbl>, char\_13 <dbl>,  
## # char\_14 <dbl>, char\_15 <dbl>, char\_16 <dbl>, char\_17 <dbl>, char\_18 <dbl>,  
## # char\_19 <dbl>, char\_20 <dbl>, char\_21 <dbl>, char\_22 <dbl>, char\_23 <dbl>,  
## # char\_24 <dbl>, char\_25 <dbl>, char\_26 <dbl>, char\_27 <dbl>, char\_28 <dbl>,  
## # char\_29 <dbl>, char\_30 <dbl>, char\_31 <dbl>, char\_32 <dbl>, char\_33 <dbl>,  
## # char\_34 <dbl>, char\_35 <dbl>, char\_36 <dbl>, char\_37 <dbl>, char\_38 <dbl>

## Joining test dataset with people dataset

test = test %>% left\_join(people , by = c('people\_id' ='people\_id'))

## Applying Naive Bayes Algorithm

require(e1071)

## Loading required package: e1071

model = naiveBayes(maindata2[,-15],as.factor(maindata2$outcome),laplace = 1)

predict = predict(model, newdata = test)

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_10.y'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_11'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_12'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_13'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_14'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_15'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_16'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_17'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_18'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_19'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_20'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_21'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_22'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_23'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_24'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_25'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_26'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_27'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_28'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_29'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_30'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_31'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_32'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_33'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_34'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_35'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_36'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(model, newdata = test): Type mismatch between  
## training and new data for variable 'char\_37'. Did you use factors with numeric  
## labels for training, and numeric values for new data?

df<-data.frame(predict)  
df%>%head(10)

## predict  
## 1 0  
## 2 0  
## 3 1  
## 4 1  
## 5 1  
## 6 1  
## 7 1  
## 8 1  
## 9 1  
## 10 1

## Converting the output into a csv file

output=as.data.frame(test$activity\_id)  
output=output%>%mutate(activity\_id=test$activity\_id, outcome=predict)  
write.csv(output,file="KaggleSubmission13.csv",row.names=FALSE)