Group_3_Analysis

Group_3

2023-03-10

Aim of Analysis

Exploratory Analysis

Data Wrangling

```
imdb <- read.csv("https://raw.githubusercontent.com/fanghuaqiu/DMML-Group-3/main/group_3.csv",na.string
  mutate(ROI = (gross-budget)/budget) %>% #define ROI
  mutate(scs = ifelse(ROI>=1,1,0)) %>% #define binary labelled output
  mutate(title_year = as.factor(title_year),
         aspect_ratio = as.factor((aspect_ratio)),
         scs = as.factor(scs),
         content_rating = replace_na(content_rating, "Unknown"),
         country = str_replace(country," ","_"),
         content_rating = str_replace(content_rating," ","_"))%>%
  mutate(content_rating = str_replace(content_rating,"-1","_1"))
```

```
same_sample <- function(x,y){
  set.seed(84)
  return(sample(x,y))
}
index_test <- same_sample(1:nrow(imdb), round(0.25*nrow(imdb)))
index_val <- same_sample((1:nrow(imdb))[-index_test], round(0.25*nrow(imdb)))
index_train <- setdiff(1:nrow(imdb),c(index_test,index_val))
test_imdb <- imdb[index_test,]
valid_imdb <- imdb[index_val,]
train_imdb <- imdb[index_train,]
# training set for tree model
index_train_t <- same_sample(setdiff(1:nrow(imdb),c(index_test,index_val)), round(0.25*nrow(imdb)))
train_imdb_t <- imdb[index_train_t,]
```

```
same_sample <- function(x,y){
  set.seed(84)
  return(sample(x,y))
}
# balanced
index_train_1 <- same_sample(which(imdb$scs == '1'), round(0.25*nrow(imdb)))
index_train_0 <- same_sample(which(imdb$scs == '0'), round(0.25*nrow(imdb)))
```

```r
index_train_b <- c(index_train_1,index_train_0)
index_val_b <- same_sample((1:nrow(imdb))[-index_train_b], round(0.25*nrow(imdb)))
index_test_b <- setdiff(1:nrow(imdb),c(index_val_b,index_train_b))
test_imdb_b <- imdb[index_test_b,]
valid_imdb_b <- imdb[index_val_b,]
train_imdb_b <- imdb[index_train_b,]
dim(test_imdb_b)
```

```
## [1] 250  30
```

```r
# training set for tree model
index_train_t_1 <- same_sample(which(train_imdb_b$scs == '1'), round(0.25*nrow(train_imdb_b)))
index_train_t_0 <- same_sample(which(train_imdb_b$scs == '0'), round(0.25*nrow(train_imdb_b)))
index_train_b_t <- c(index_train_t_1,index_train_t_0)
train_imdb_b_t <- train_imdb_b[index_train_b_t,]
```

Modeling

KNN

LDA

Tree

SVM

Definition of SVM

SVM denotes support vector machines. The main idea of fitting a SVM is trying to find a hyperplane to separate the observations into two parts. The shortest vertical distance from a hyperplane to a point on both sides is called margin. To find an optimal SVM is to find a SVM with the biggest margin.

Data wrangling for SVM modelling

After analyzing the dataset, we found that there are many different levels in the categorical variables in the dataset. After our dataset is divided into training set, validation set, and test set, the number of samples contained in each level of the categorical variables in each set is not large, so if the model is built by including categorical variables, the accuracy of the model will decrease, so we first remove the categorical variables in the dataset. At the same time, because there may be an inclusion relationship between some variables, such as the number of Facebook likes of the movie, only some important numeric variables are retained, and the reserved numeric variables are shown in table 1.

Table 1: The table of variables used in SVM and their data types.

Varible	Type
duration	numeric
director_facebook_likes	numeric
num_critic_for_reviews	numeric
num_user_for_reviews	numeric

Varible	Type
num_voted_users	numeric
cast_total_facebook_likes	numeric
movie_facebook_likes	numeric
imdb_score	numeric
scs	factor

After selecting the variables that need to be used for modeling, we need to standardize the numerical variables and unify the dimensions of the numerical variables to meet the modeling requirements.

SVM Model

We use radial, polynomial, and linear as kernels for the SVM model. At the same time, a value range is determined for the parameters that need to be used in the three models. The optimal prediction model is selected by comparing the prediction accuracy of the validate set.

We can see from the table 2, the model with a polynomial kernel has the highest validation accuracy. Therefore, we choose to use the model with a polynomial kernel as the final SVM model.

Table 2: The table of the validation accuracy of the SVM model.

Linear	Polynomial	radial
0.612	0.772	0.568

Neural Network

Conclusion

In the support vector machine part, an SVM model using different parameters and kernels is established to compare and find the optimal SVM model. By comparing the validation accuracy of the validation set of SVM models of different kernels, it is found that the highest prediction accuracy can be obtained by using polynomial kernels with a degree of 3, a cost of 20, and a gamma of 0.125. We predict the test set with the following accuracy:0.744.