Predizendo o banco de dados não rotulado

Para a predição do banco de dados não rotulado, escolheu-se o modelo XGBoost para cada tag (com o número de árvores igual a 100), isto é o modelo 10. Em relação ao banco de dados, utilizou-se aquele com a matriz documento texto em que havia o título e a sinopse juntos, pegando apenas as palavras que apareciam mais de 5% das vezes.

Mas dessa vez, para o ajuste do modelo, utilizou-se 100% do conjunto de dados rotulados, de forma a obter o máximo de informação possível.

Com base nessa matriz documento termo do treino, criou-se outra matriz documento termo, mas agora para os dados não rotulados. Dessa forma, o dtm dos dados não rotulados possuirá apenas as palavras com quais o modelo foi treinado.

```
## <<DocumentTermMatrix (documents: 9828, terms: 1033)>>
## Non-/sparse entries: 1302618/8849706
## Sparsity           : 87%
## Maximal term length: 12
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

Desse modo, nota-se que com o dtm2 (conjunto rotulado) possui uma esparcidade razoável e uma boa quantidade de termos e filmes.

Cria-se o dtm do conjunto de dados não rotulado.

```
resumo = VCorpus(VectorSource(c(paste(unlabeled_data$title,
                                      unlabeled_data$plot_synopsis))),
                 readerControl = list(languague = "en"))

resumo1 =  tm_map(resumo,
                  removeWords,
                  stopwords(language = "en",source = "smart"))

dtm_teste_unlabeled = resumo1 %>%
  DocumentTermMatrix(control = list(tolower=T,
                                    removePunctuation = T,
                                    removeNumbers = T,
                                    stripWhitespace = T,
                                    stopwords = T,
                                    stemming = T,
                                    weighting= weightTfIdf,
                                    ### Pega-se apenas as palavras do rotulado
                                    dictionary=Terms(dtm2)))
dtm_teste_unlabeled
```

```
## <<DocumentTermMatrix (documents: 5000, terms: 1033)>>
## Non-/sparse entries: 664021/4500979
## Sparsity           : 87%
## Maximal term length: 12
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

Nota-se novamente uma esparsidade razoável. Mas agora há 5000 filmes (número de filmes dos dados não rotulados), e com as mesmas palavras (covariáveis), que foram 1033. Em seguida, treina-se o modelo escolhido.

```
lista_modelos_xgb_final = list()

for(i in 1:ncol(dados[,-83]) ){

  xgb_mat_treino_tudo =
    xgb.DMatrix(as.matrix(dtm2),
                label = as.matrix(dados[,i]))

  set.seed(i+500)
  #cat('A coluna eh ',i,'\n')

  modelo_xgb =
  xgboost(data = xgb_mat_treino_tudo,
     nrounds = 100,
     eval_metric = "auc",
     verbose = 0,
     objective = "binary:logistic")

  lista_modelos_xgb_final[[i]] = modelo_xgb
}
```

Com os micro modelos XGboost para cada tag criados, basta predizer as probabilidades no banco de dados não rotulado.

```
mat_prob_unlabeled = matrix(NA,
                            nrow=nrow(dtm_teste_unlabeled),
                            ncol = ncol(dados[,-83]) )

for(i in 1:ncol(dados[,-83]) ){

  mat_prob_unlabeled[,i] <-
    predict(lista_modelos_xgb_final[[i]],
            xgb.DMatrix(as.matrix(dtm_teste_unlabeled)),
            ype="prob")

}
```

Com isso, tem-se uma matriz que representa as probabilidades de cada filme possuir determinadas tags. Para continuar a análise, é preciso estabelecer um ponto de corte de modo a obter 1 (se a probabilidade for maior) e 0 (se a probabilidade for menor).

Para tal, escolhe-se a proporção de tags no banco de dados rotulado (semelhante ao ponto de corte usado para a métrica de F1 dos modelos anteriores, sendo tal métrica baseada na proporção das tags).

```
unlabeled_data_predito = unlabeled_data

unlabeled_data_predito[,3:(ncol(dados[,-83])+2)] =
  ifelse(mat_prob_unlabeled>
           matrix(rep(colMeans(dados[,-83]),nrow(dtm_teste_unlabeled)),
                  byrow = T,
                  ncol = ncol(dados[,-83]),
                  nrow = nrow(dtm_teste_unlabeled) ),
         1,0)
```

```
unlabeled_data_predito[1:5,1:5]
```

```
## # A tibble: 5 x 5
##   title                 plot_synopsis                          cult horror gothic
##   <chr>                 <chr>                                 <dbl>  <dbl>  <dbl>
## 1 Laws of Attraction    "High-powered divorce attorneys Aud~     0      0      0
## 2 George and the Dragon "Note: Significant plot details fol~     1      0      0
## 3 The Ref               "In a charming Connecticut village,~     0      0      0
## 4 Teenage Cave Man      "The movie opens with black and whi~     1      0      1
## 5 A Tale of Two Cities  "=== Book the First: Recalled to Li~     0      0      0
```

Com isso, tem-se os dados não rotulados agora rotulados, com valores 1 (se tag está presente) e 0 (se tag não está presente). Basta agora exportar tal tabela como um CSV.

```
library(readr)
write_csv(unlabeled_data_predito,file="LubenLuizVinicius.csv")
```