Bike sharing

Load default libraries

```r
library(data.table)
library(dplyr)
library(tm)
library(ggplot2)
library(caret)
library(stringi)
library(FeatureHashing)
library(xgboost)
theme_set(theme_bw())
set.seed(42)
```

Read the data

```r
train = fread("./prepared_train.csv", header = T, sep = ",", integer64 = "numeric")
validate = fread("./prepared_validate.csv", header = T, sep = ",", integer64 = "numeric")
test = fread("./prepared_test.csv", header = T, sep = ",", integer64 = "numeric")

test2 = fread("./test.csv", header = T, sep = ",", integer64 = "numeric")
```

Change type to categorical variables

```r
fix_types = function(data) {
  data %>% mutate(
  month = as.character(month),
  day = as.character(day),
  weekday = as.character(weekday),
  hour = as.character(hour),
  season = as.character(season),
  holiday = as.character(holiday),
  workingday = as.character(workingday),
  weather = as.character(weather))
}
train = fix_types(train)
validate = fix_types(validate)
test = fix_types(test)
```

Prepare features

Prepare target

We going to predict log(Y+1) to optimize the target cost function

```r
preparedTrainTarget = log(train$count + 1)
preparedValidateTarget = log(validate$count + 1)
preparedFullTarget =c(preparedTrainTarget, preparedValidateTarget)
```

Xgboost train and cross-validate

```r
dtrain <- xgb.DMatrix(train_matrix, label = preparedTrainTarget)

model = xgboost(dtrain, nround=100, nthread = 2, nfold = 5, metrics=list("rmse"),
                  max.depth = 3, eta = 0.1, objective = "reg:linear", print.every.n = 33)
```

```
## [0]  train-rmse:3.910831
## [33] train-rmse:0.755147
## [66] train-rmse:0.603379
## [99] train-rmse:0.529682
```

```r
history <- xgb.cv(data = dtrain, nround=150, nthread = 2, nfold = 5, metrics=list("rmse"),
                  max.depth = 3, eta = 0.1, objective = "reg:linear", print.every.n = 33)
```

```
## [0]  train-rmse:3.910832+0.005531    test-rmse:3.910809+0.025410
## [33] train-rmse:0.750372+0.004117    test-rmse:0.762903+0.011960
## [66] train-rmse:0.603587+0.006684    test-rmse:0.621172+0.014271
## [99] train-rmse:0.532241+0.010550    test-rmse:0.552558+0.012204
## [132]    train-rmse:0.492933+0.008041    test-rmse:0.516385+0.012395
```

```r
print(tail(history))
```

```
##    train.rmse.mean train.rmse.std test.rmse.mean test.rmse.std
## 1:        0.482876       0.007863       0.507664      0.012382
## 2:        0.482116       0.007546       0.506943      0.012814
## 3:        0.481485       0.007599       0.506411      0.012642
## 4:        0.480852       0.007480       0.505880      0.012897
## 5:        0.480169       0.007492       0.505234      0.013035
## 6:        0.479464       0.007735       0.504545      0.012662
```

Validate Score

```r
validate_predictions = predict(model, validate_matrix)
RMSE(validate_predictions, preparedValidateTarget)
```

```
## [1] 0.5618725
```

Train on full dataset

```r
dtrain_final <- xgb.DMatrix(full_train_matrix, label = preparedFullTarget)

model_final = xgboost(dtrain_final, nround=100, nthread = 2, nfold = 5, metrics=list("rmse"),
                  max.depth = 3, eta = 0.1, objective = "reg:linear", print.every.n = 33)
```

```
## [0]  train-rmse:3.926399
## [33] train-rmse:0.759091
## [66] train-rmse:0.609704
## [99] train-rmse:0.544402
```

```r
history <- xgb.cv(data = dtrain_final, nround=100, nthread = 2, nfold = 5, metrics=list("rmse"),
                  max.depth = 3, eta = 0.1, objective = "reg:linear", print.every.n = 33)
```

```
## [0]  train-rmse:3.926431+0.005475    test-rmse:3.926381+0.024544
## [33] train-rmse:0.755690+0.006431    test-rmse:0.762112+0.017506
## [66] train-rmse:0.608737+0.005373    test-rmse:0.618299+0.018470
## [99] train-rmse:0.546318+0.007077    test-rmse:0.558473+0.021436
```

Predict and un-log predictions

```r
predictions = predict(model_final, test_matrix)
fixed_predictions = exp(predictions) - 1
```

Write the result

```r
result = cbind(test2$datetime, fixed_predictions) %>% as.data.frame()
names(result) = c('datetime', 'count')
write.csv(result, 'submission.csv', quote = F, row.names = F)
```