FinalProject

Abhishek Shah

4/12/2022

R Markdown

Libraries

library(dplyr) library(lubridate) library(ggplot2) library(reshape) library(gbm)

Assigning and Cleaning/ Pre - Processing Data

train <- read.csv(“sales_train.csv”) items <- read.csv(“items.csv”) test <- read.csv(“test.csv”)

merged.train <- merge.data.frame(train, items, by = c(“item_id”)) merged.train $item_name < -NULL head(merged.train) str(merged.train) rm(train) rm(items) train < -as.data.frame(merged.train) train$ d <- dmy(train$date)

Feature Engineering

train$year < -year(train$date) train$month < -month(train$date) train$day < -day(train$date) train $weekday < -weekdays(train$date) train$year < -as.factor(train$year) train$weekday < -as.factor(train$weekday) train $month < -as.factor(train$month) train$day < -as.factor(train$day)

cnt_month <- train %>% group_by(year, month, shop_id, item_id) %>% summarise(item_cnt_month = sum(item_cnt_day)) %>% ungroup() train <- train %>% left_join(cnt_month, by = c(“year”, “month”, “shop_id”, “item_id”)) rm(cnt_month)

Assigning new data and evaluating correlation

summary(train) colSums(is.na(train)) num.cols <- sapply(train, is.numeric) train_numcols <- train[, num.cols] train_numcols $date_block_num < -NULL train_numcols$month <- NULL train_numcols$day <- NULL cor(train_numcols) correlation = melt(cor(train_numcols))

ggplot(data = correlation, aes(x = X1, y = X2, fill = value))+ geom_tile()+ scale_fill_gradient(low=“grey”,high=“darkred”)+ geom_text(aes(x = X1, y = X2, label = round(value,2)),size=4)+ labs(title = “Correlation Matrix”, x = “Numeric Column(s)”, y = “Numeric Column(s)”, fill = “Coefficient Range”) + theme(axis.text.x=element_text(angle=45, vjust=0.5))

Exploring data and Visualizing

plotting and ranking shop sale

shop_sale <- train %>% select(shop_id, item_cnt_day) %>% group_by(shop_id) %>% summarise(item_cnt_day = sum(item_cnt_day, na.rm = TRUE))

ggplot(data = shop_sale, mapping = aes(x = reorder(shop_id, item_cnt_day), y = item_cnt_day, fill = factor(shop_id))) + geom_histogram(stat = “identity”) + xlab(“Shop ID”) + ylab(“Sales Count”)+ ggtitle(label = “Shop sales”)

plotting category sales

ctgry_sale <- train %>% select(item_category_id, item_cnt_day) %>% group_by(item_category_id) %>% summarise(item_cnt_day = sum(item_cnt_day, na.rm = TRUE))

ggplot(data = ctgry_sale, mapping = aes(x = reorder(item_category_id,item_cnt_day), y = item_cnt_day, fill = factor(item_category_id))) + geom_histogram(stat = “identity”) + coord_flip() + xlab(“Item Category”) + ylab(“Sales Count”) + ggtitle(“Sale Item Category wise”)

plotting best selling items

best_selling_items <- train %>% group_by(item_category_id) %>% summarise(total_gross = sum(item_cnt_day * item_price)) %>% arrange(desc(total_gross)) ggplot(best_selling_items, aes(x = reorder(item_category_id, total_gross), y = total_gross, fill = factor(item_category_id))) + geom_histogram(stat = “identity”) + xlab(“Category ID”) + ylab(“Total Gross”)+ ggtitle(“Total Gross per Item category”) + coord_flip()

Modelling data using General Boosting Machine

gbm_model = gbm(item_cnt_day ~ shop_id + item_id, data = train, shrinkage = 0.01, distribution = “gaussian”, n.trees = 5000, interaction.depth = 3, bag.fraction = 0.7, train.fraction = 0.8, cv.folds = 5, n.cores = NULL, verbose = T)

Predicting

result = predict(gbm_model,newdata = test[,c(“shop_id”,“item_id”)], n.trees = 5000)

sqrt(min(gbm_model$cv.error)) gbm.perf(gbm_model, method = “cv”)

Framing the data as per the requirements

submission = data.frame(ID = test$ID, item_cnt_month = result)

write.csv(submission, “submission.csv”, row.names = F)