Group10FinalProjectRMD

install.packages("sparklyr")  
install.packages("aws.s3")  
install.packages("tidyverse")  
install.packages("dplyr")  
library(sparklyr)  
library(aws.s3)  
library(tidyverse)  
library(dplyr)

config <- spark\_config()  
config[["sparklyr.shell.driver-memory"]] <- "10G"  
sc <- spark\_connect(master = "yarn",  
 config = config,  
 spark\_home = "/usr/lib/spark")

## Creativity Part

Here, we are loading our data for our **green taxi** cab analysis. The entire dataset is a combined file of every month (January through December) for the years 2017 and 2019. In this section, we’re also splitting our data into train and test sets and are also doing a bit of data cleaning as well, such as picking specific variables for our analysis and removing NA values.

start = proc.time()  
Taxi\_green\_tbl = spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Green/Green2.csv", infer\_schema = TRUE, header = TRUE)  
proc.time() - start  
  
#sdf\_describe(Taxi\_green\_tbl)  
  
partitions <- Taxi\_green\_tbl %>%  
 sdf\_random\_split(training = 0.1, test = 0.9, seed = 1111)  
  
Taxi\_green\_training <- partitions$training   
Taxi\_green\_test <- partitions$test  
  
#sdf\_describe(Taxi\_green\_training)  
  
#Pick variables we're interested in (apply to both training and test sets)  
Taxi\_green\_training = Taxi\_green\_training %>% select(passenger\_count,trip\_distance,fare\_amount,tip\_amount,total\_amount,payment\_type)  
Taxi\_green\_test = Taxi\_green\_test %>% select(passenger\_count,trip\_distance,fare\_amount,tip\_amount,total\_amount,payment\_type)  
  
#Remove all NA values for both training and test  
Taxi\_green\_training = na.omit(Taxi\_green\_training)  
Taxi\_green\_test = na.omit(Taxi\_green\_test)

Here we are trying to find the correlation between passenger count, trip distance, and total amount for the green cab data.

ml\_corr(Taxi\_green\_training, columns = c("passenger\_count", "trip\_distance", "total\_amount"))

We will further inspect our variables for our green cab data.

summary(Taxi\_green\_training %>%   
 select(passenger\_count) %>%  
 collect())  
  
summary(Taxi\_green\_training %>%   
 select(trip\_distance) %>%  
 collect())  
  
Taxi1\_tbl = Taxi\_green\_training %>%  
 select(payment\_type, total\_amount) %>%  
 collect()  
summary(Taxi1\_tbl$total\_amount)  
summary(as.factor(Taxi1\_tbl$payment\_type))

Here, we will be plotting our clean data of “Trip Distance v. Passenger Count” for our green cab data.

start = proc.time()  
Taxi\_green\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 10 & trip\_distance < 100) %>%  
 collect() %>%  
 ggplot(aes(x = passenger\_count, y = trip\_distance)) +  
 geom\_point() +   
 ggtitle("Trip Distance v. Passenger Count") +  
 xlab("Passenger Count") +  
 ylab("Trip Distance")  
proc.time() - start

We will further investigate passenger count and trip distance.

summary(Taxi\_green\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 100 & trip\_distance < 100) %>%  
 collect())

We will build two linear regression models, one to see how passenger count affects trip distance, and the other to see how passenger count affects total amount.

# Model 1  
fitlm = Taxi\_green\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 100 & trip\_distance < 100) %>%  
 ml\_linear\_regression(trip\_distance~passenger\_count)  
  
tidy(fitlm)  
pred = ml\_predict(fitlm,Taxi\_green\_test)  
ml\_regression\_evaluator(pred, label\_col = "trip\_distance", metric\_name = "rmse")  
  
  
# Model 2  
fitlm2 = Taxi\_green\_training %>%   
 select(passenger\_count,total\_amount) %>%  
 filter(passenger\_count > 0 & total\_amount > 0) %>%  
 filter(passenger\_count < 10 & total\_amount < 500) %>%  
 ml\_linear\_regression(total\_amount~passenger\_count)  
  
tidy(fitlm2) #paremeter estimate table  
pred = ml\_predict(fitlm2,Taxi\_green\_test)  
ml\_regression\_evaluator(pred, label\_col = "total\_amount", metric\_name = "rmse")

We are building a logistic regression model to predict payment type from total amount.

fitlog <- Taxi\_green\_training %>% select(payment\_type,total\_amount) %>%   
 filter(payment\_type == 1 | payment\_type == 2) %>%  
 ml\_logistic\_regression(payment\_type~total\_amount)  
  
fitlog  
summary(fitlog)  
tidy(fitlog)  
  
pred <- ml\_predict(fitlog, Taxi\_green\_test)  
ml\_binary\_classification\_evaluator(pred)  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "f1")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "accuracy")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "weightedPrecision")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "weightedRecall")

Here, we are loading our data for our **yellow taxi** cab analysis. The entire dataset is a combined file of the months May and June for the years 2015, 2017, 2019, and 2020. Like before, in this section, we’re splitting our data into train and test sets and are also doing a bit of data cleaning as well, such as picking specific variables for our analysis and removing NA values.

start = proc.time()  
Taxi\_yellow\_tbl = spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Yellow/Yellow.csv", infer\_schema = TRUE, header = TRUE)  
proc.time() - start  
  
#sdf\_describe(Taxi\_yellow\_tbl)  
#head(Taxi\_yellow\_tbl)  
  
partitions <- Taxi\_yellow\_tbl %>%  
 sdf\_random\_split(training = 0.1, test = 0.9, seed = 1111)  
  
Taxi\_yellow\_training <- partitions$training   
Taxi\_yellow\_test <- partitions$test  
  
#Pick variables we're interested in  
Taxi\_yellow\_training = Taxi\_yellow\_training %>% select(passenger\_count,trip\_distance,fare\_amount,tip\_amount,total\_amount,payment\_type)  
Taxi\_yellow\_test = Taxi\_yellow\_test %>% select(passenger\_count,trip\_distance,fare\_amount,tip\_amount,total\_amount,payment\_type)  
  
#Remove NA values  
Taxi\_yellow\_training = na.omit(Taxi\_yellow\_training)  
Taxi\_yellow\_test = na.omit(Taxi\_yellow\_test)

Here we are trying to find the correlation between passenger count, trip distance, and total amount for the yellow cab data.

ml\_corr(Taxi\_yellow\_training, columns = c("passenger\_count", "trip\_distance", "total\_amount"))

We will further inspect our variables for our yellow cab data.

summary(Taxi\_yellow\_training %>%   
 select(passenger\_count) %>%  
 collect())  
  
summary(Taxi\_yellow\_training %>%   
 select(trip\_distance) %>%  
 collect())  
  
Taxi1\_tbl = Taxi\_yellow\_training %>%  
 select(payment\_type, total\_amount) %>%  
 collect()  
  
summary(Taxi1\_tbl$total\_amount)  
summary(factor(Taxi1\_tbl$payment\_type))

Here, we will be plotting our clean data of “Trip Distance v. Passenger Count” for our yellow cab data.

start = proc.time()  
Taxi\_yellow\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 10 & trip\_distance < 100) %>%  
 collect() %>%  
 ggplot(aes(x = passenger\_count, y = trip\_distance)) +  
 geom\_point() +   
 ggtitle("Trip Distance v. Passenger Count") +  
 xlab("Passenger Count") +  
 ylab("Trip Distance")  
proc.time() - start

We will further investigate passenger count and trip distance.

summary(Taxi\_yellow\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 100 & trip\_distance < 100) %>%  
 collect())

We will again build two linear regression models, but this time for our yellow cab data.

# Model 1  
fitlm = Taxi\_yellow\_training %>%   
 select(passenger\_count,trip\_distance) %>%  
 filter(passenger\_count > 0 & trip\_distance > 0) %>%  
 filter(passenger\_count < 100 & trip\_distance < 100) %>%  
 ml\_linear\_regression(trip\_distance~passenger\_count)  
tidy(fitlm)  
  
# Model 2  
fitlm2 = Taxi\_yellow\_training %>%   
 select(passenger\_count,total\_amount) %>%  
 filter(passenger\_count > 0 & total\_amount > 0) %>%  
 filter(passenger\_count < 10 & total\_amount < 500) %>%  
 ml\_linear\_regression(total\_amount~passenger\_count)  
tidy(fitlm2)

We are building a logistic regression model to predict payment type from total amount.

fitlog <- Taxi\_yellow\_training %>% select(payment\_type,total\_amount) %>%   
 filter(payment\_type == 1 | payment\_type == 2) %>%  
 ml\_logistic\_regression(payment\_type~total\_amount)  
  
fitlog  
summary(fitlog)  
tidy(fitlog)  
  
pred <- ml\_predict(fitlog, Taxi\_yellow\_test)  
ml\_binary\_classification\_evaluator(pred)  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "f1")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "accuracy")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "weightedPrecision")  
ml\_multiclass\_classification\_evaluator(pred, metric\_name = "weightedRecall")

Here we are loading our for-hire groups data, partitioning into a train and test split, doing some variable manipulation, and creating bargraphs and plots.

forhire = spark\_read\_csv(sc,path = "s3://stat6306studentfilebucket/Andreas and Webb/ForHireTaxi/ForHireVehicle2019.csv", infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
#sdf\_describe(forhire)  
  
  
partitions <- forhire %>% sdf\_random\_split(training = 0.99,test = 0.01, seed = 1)  
forhirebig <- partitions$training  
forhiresmall <- partitions$test  
  
#sdf\_describe(forhiresmall)  
  
forhiresmall1 <- forhiresmall %>% select(hvfhs\_license\_num) %>% collect()  
  
forhiresmall2 <- forhiresmall1 %>% mutate(hvfhs\_license\_num = if\_else(hvfhs\_license\_num == "HV0002","Juno",hvfhs\_license\_num)) %>%  
 mutate(hvfhs\_license\_num = if\_else(hvfhs\_license\_num == "HV0003","Uber",hvfhs\_license\_num)) %>%  
 mutate(hvfhs\_license\_num = if\_else(hvfhs\_license\_num == "HV0004","Via",hvfhs\_license\_num)) %>%  
 mutate(hvfhs\_license\_num = if\_else(hvfhs\_license\_num == "HV0005","Lyft",hvfhs\_license\_num))  
  
barplot((summary(factor(forhiresmall2$hvfhs\_license\_num))),main="Histogram of For Hire Rides",   
 xlab="For Hire Ride Service",  
 ylab = "Count",  
 border="black",   
 col="gray")  
  
forhiresmall3 <- forhiresmall %>% select(SR\_Flag) %>% collect()  
  
forhiresmall4 <- forhiresmall3 %>% mutate(SR\_Flag = if\_else(SR\_Flag == 1, "Shared Ride","Non Shared Ride"))  
  
barplot((summary(factor(forhiresmall4$SR\_Flag))),  
 main = "Percentage of Shared Rides",  
 xlab = "Shared Ride or Non Shared (NA)",  
 ylab = "Count",  
 border = "black",  
 col = "gray")

## Prediction Part

This is the **prediction part** of our analysis. We will be predicting tip amount with yellow cab data using linear regression and payment type with green cab data using logistic regression.

We will use trip distance and payment type as our two predictors to predict tip amount for our yellow taxi cab data.

#Yellow Predictions 2009  
yellow2009train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Yellow/yellow\_2009-03.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2009test <- spark\_read\_csv(sc, path = "s3://stat6306project/Yellow Test Student/TaxiTest2009\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2009sub <- yellow2009train %>% select(Tip\_Amt,Trip\_Distance,Passenger\_Count,Payment\_Type) %>% filter(Payment\_Type == "CREDIT")  
yellow2009subtest <- yellow2009test %>% select(Trip\_Distance,Passenger\_Count,Payment\_Type)  
  
fityellow2009 <- yellow2009sub %>% ml\_linear\_regression(Tip\_Amt~Trip\_Distance+Passenger\_Count)  
tidy(fityellow2009)  
  
preds2009 <- ml\_predict(fityellow2009,yellow2009subtest)  
  
preds2009subset <- preds2009 %>% select(prediction) %>% collect()  
preds2009subset1 <- yellow2009test %>% select(ID) %>% collect()  
  
predictions2009 <- matrix(0,nrow = 32863,ncol = 2)  
colnames(predictions2009) <- c("ID","tip\_amount")  
  
predictions2009[,1] <- preds2009subset1$ID  
  
predictions2009[,2] <- preds2009subset$prediction  
  
#Yellow Predictions 2011  
yellow2011train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Yellow/yellow\_2011-02.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2011test <- spark\_read\_csv(sc, path = "s3://stat6306project/Yellow Test Student/TaxiTest2011\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2011sub <- yellow2011train %>% select(tip\_amount,trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == "CRD")  
yellow2011subtest <- yellow2011test %>% select(trip\_distance,passenger\_count,payment\_type)  
  
fityellow2011 <- yellow2011sub %>% ml\_linear\_regression(tip\_amount~trip\_distance+passenger\_count)  
tidy(fityellow2011)  
  
preds2011 <- ml\_predict(fityellow2011,yellow2011subtest)  
  
preds2011subset <- preds2011 %>% select(prediction) %>% collect()  
preds2011subset1 <- yellow2011test %>% select(ID) %>% collect()  
  
predictions2011 <- matrix(0,nrow = 61535,ncol = 2)  
colnames(predictions2011) <- c("ID","tip\_amount")  
  
predictions2011[,1] <- preds2011subset1$ID  
  
predictions2011[,2] <- preds2011subset$prediction  
  
#Yellow Predictions 2018  
yellow2018train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Yellow/yellow\_tripdata\_2018-03.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2018test <- spark\_read\_csv(sc, path = "s3://stat6306project/Yellow Test Student/TaxiTest2018\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2018sub <- yellow2018train %>% select(tip\_amount,trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == 1)  
yellow2018subtest <- yellow2018test %>% select(trip\_distance,passenger\_count,payment\_type)  
  
fityellow2018 <- yellow2018sub %>% ml\_linear\_regression(tip\_amount~trip\_distance+passenger\_count)  
tidy(fityellow2018)  
  
preds2018 <- ml\_predict(fityellow2018,yellow2018subtest)  
  
preds2018subset <- preds2018 %>% select(prediction) %>% collect()  
preds2018subset1 <- yellow2018test %>% select(ID) %>% collect()  
  
predictions2018 <- matrix(0,nrow = 66530,ncol = 2)  
colnames(predictions2018) <- c("ID","tip\_amount")  
  
predictions2018[,1] <- preds2018subset1$ID  
  
predictions2018[,2] <- preds2018subset$prediction  
  
#Yellow Predictions 2020  
yellow2020train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Yellow/yellow\_tripdata\_2020-05.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2020test <- spark\_read\_csv(sc, path = "s3://stat6306project/Yellow Test Student/TaxiTest2020\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
yellow2020sub <- yellow2020train %>% select(tip\_amount,trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == 1)  
yellow2020subtest <- yellow2020test %>% select(trip\_distance,passenger\_count,payment\_type)  
  
fityellow2020 <- yellow2020sub %>% ml\_linear\_regression(tip\_amount~trip\_distance+passenger\_count)  
tidy(fityellow2020)  
  
preds2020 <- ml\_predict(fityellow2020,yellow2020subtest)  
  
preds2020subset <- preds2020 %>% select(prediction) %>% collect()  
preds2020subset1 <- yellow2020test %>% select(ID) %>% collect()  
  
predictions2020 <- matrix(0,nrow = 70027,ncol = 2)  
colnames(predictions2020) <- c("ID","tip\_amount")  
  
predictions2020[,1] <- preds2020subset1$ID  
  
predictions2020[,2] <- preds2020subset$prediction  
  
  
#Put everything together  
  
yellowpredictfinal <- rbind(predictions2009,predictions2011,predictions2018,predictions2020)  
  
Sys.setenv("AWS\_ACCESS\_KEY\_ID" = "AKIAJFPIUJBDHKKBR6SA",  
 "AWS\_SECRET\_ACCESS\_KEY" = "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",  
 "AWS\_DEFAULT\_REGION" = "us-east-2")  
  
aws.s3::bucketlist()  
aws.s3::get\_bucket("ds6306unit13")  
  
#s3write\_using(yellowpredictfinal, FUN = write.csv,bucket = "stat6306studentfilebucket",object = "Group10YellowTaxiFinalPredictions")

Next, we will be using trip distance and passenger count to predict payment type (credit or cash) for our green taxi cab data.

#Green Predictions 2015  
  
green2015train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Green/Green 2015/green\_tripdata\_2015-11.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
green2015test <- spark\_read\_csv(sc, path = "s3://stat6306project/Green Test Student/TaxiTest2015G\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
green2015sub <- green2015train %>% select(Trip\_distance,Passenger\_count,Payment\_type) %>% filter(Payment\_type == 1 | Payment\_type == 2)  
green2015subtest <- green2015test %>% select(Trip\_distance,Passenger\_count)  
  
fitgreen2015 <- green2015sub %>% ml\_logistic\_regression(Payment\_type~Trip\_distance+Passenger\_count)  
tidy(fitgreen2015)  
  
preds2015 <- ml\_predict(fitgreen2015,green2015subtest)  
  
preds2015subset <- preds2015 %>% select(prediction) %>% collect()  
preds2015subset1 <- green2015test %>% select(ID) %>% collect()  
  
predictions2015 <- matrix(0,nrow = 76323,ncol = 2)  
colnames(predictions2015) <- c("ID","tip\_amount")  
  
predictions2015[,1] <- preds2015subset1$ID  
  
predictions2015[,2] <- preds2015subset$prediction  
  
  
#Green Predictions 2017  
  
green2017train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Green/Green 2017/green\_tripdata\_2017-04.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
green2017test <- spark\_read\_csv(sc, path = "s3://stat6306project/Green Test Student/TaxiTest2017G\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
green2017sub <- green2017train %>% select(trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == 1 | payment\_type == 2)  
green2017subtest <- green2017test %>% select(trip\_distance,passenger\_count)  
  
fitgreen2017 <- green2017sub %>% ml\_logistic\_regression(payment\_type~trip\_distance+passenger\_count)  
tidy(fitgreen2017)  
  
preds2017 <- ml\_predict(fitgreen2017,green2017subtest)  
  
preds2017subset <- preds2017 %>% select(prediction) %>% collect()  
preds2017subset1 <- green2017test %>% select(ID) %>% collect()  
  
predictions2017 <- matrix(0,nrow = 53967,ncol = 2)  
colnames(predictions2017) <- c("ID","tip\_amount")  
  
predictions2017[,1] <- preds2017subset1$ID  
  
predictions2017[,2] <- preds2017subset$prediction  
  
  
#Green Predictions 2019  
  
green2019train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Green/Green 2019/green\_tripdata\_2019-06.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
green2019test <- spark\_read\_csv(sc, path = "s3://stat6306project/Green Test Student/TaxiTest2019G\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
green2019sub <- green2019train %>% select(trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == 1 | payment\_type == 2)  
green2019subtest <- green2019test %>% select(trip\_distance,passenger\_count)  
  
fitgreen2019 <- green2019sub %>% ml\_logistic\_regression(payment\_type~trip\_distance+passenger\_count)  
tidy(fitgreen2019)  
  
preds2019 <- ml\_predict(fitgreen2019,green2019subtest)  
  
preds2019subset <- preds2019 %>% select(prediction) %>% collect()  
preds2019subset1 <- green2019test %>% select(ID) %>% collect()  
  
predictions2019 <- matrix(0,nrow = 23395,ncol = 2)  
colnames(predictions2019) <- c("ID","tip\_amount")  
  
predictions2019[,1] <- preds2019subset1$ID  
  
predictions2019[,2] <- preds2019subset$prediction  
  
  
#Green Predictions 2020  
  
green2020train <- spark\_read\_csv(sc, path = "s3://stat6306studentfilebucket/Andreas and Webb/Taxi/Green/Green 2020/green\_tripdata\_2020-05.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
green2020test <- spark\_read\_csv(sc, path = "s3://stat6306project/Green Test Student/TaxiTest2020G\_WO\_df.csv",infer\_schema = TRUE, header = TRUE, repartition = 100)  
  
green2020sub <- green2020train %>% select(trip\_distance,passenger\_count,payment\_type) %>% filter(payment\_type == 1 | payment\_type == 2)  
green2020subtest <- green2020test %>% select(trip\_distance,passenger\_count)  
  
fitgreen2020 <- green2020sub %>% ml\_logistic\_regression(payment\_type~trip\_distance+passenger\_count)  
tidy(fitgreen2020)  
  
predsgreen2020 <- ml\_predict(fitgreen2020,green2020subtest)  
  
predsgreen2020subset <- predsgreen2020 %>% select(prediction) %>% collect()  
predsgreen2020subset1 <- green2020test %>% select(ID) %>% collect()  
  
predictionsgreen2020 <- matrix(0,nrow = 15540,ncol = 2)  
colnames(predictionsgreen2020) <- c("ID","tip\_amount")  
  
predictionsgreen2020[,1] <- predsgreen2020subset1$ID  
  
predictionsgreen2020[,2] <- predsgreen2020subset$prediction  
  
  
#Put everything together  
  
greenpredictfinal <- rbind(predictions2015,predictions2017,predictions2019,predictionsgreen2020)  
  
Sys.setenv("AWS\_ACCESS\_KEY\_ID" = "AKIAJFPIUJBDHKKBR6SA",  
 "AWS\_SECRET\_ACCESS\_KEY" = "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",  
 "AWS\_DEFAULT\_REGION" = "us-east-2")  
  
  
aws.s3::bucketlist()  
aws.s3::get\_bucket("ds6306unit13")  
  
#s3write\_using(greenpredictfinal, FUN = write.csv,bucket = "stat6306studentfilebucket",object = "Group10GreenTaxiFinalPredictions")