گزارش پروژه پوشش جنگلی

حامد زارعی

نام شرکت: فرابرد شبکه

نام مسئول: سید محمد غفاریان

پیش بینی پوشش جنگلی

فهرست مطالب

۱. شرح مسئله

در این چالش از ما خواسته شده تا نوع پوشش جنگلی که در ۷ دسته تقسیم بندی شده‌اند را پیش‌بینی کنیم.

داده‌های موجود در داده‌های آموزش از زیر نظر گرفتن یک بخش ۳۰ × ۳۰ از ۴ ناحیه صحرایی قرار گرفته در جنگل Roosevelt National در شمال Colorado بدست آماده است.

که تعدادی از داده‌ها در شکل زیر آورده شده است:

Id	Elevation	Aspect	Slope	Horizontal_	Vertical_Di	Horizontal_	Hillshade_9	Hillshade_1	Hillshade_3	Horizontal_	Wilderness	Wilderness	Wilderness	Wilderness	Soil_Type1	Soil_Type2	Soil_Type3	Soil_Type4	Soil_Type5	Soil_Type6	So
1	2596	51	3	258	0	510	221	232	148	6279	1	0	0	0	0	0	0	0	0	0	
2	2590	56	2	212	-6	390	220	235	151	6225	1	0	0	0	0	0	0	0	0	0	
3	2804	139	9	268	65	3180	234	238	135	6121	1	0	0	0	0	0	0	0	0	0	
4	2785	155	18	242	118	3090	238	238	122	6211	1	0	0	0	0	0	0	0	0	0	
5	2595	45	2	153	-1	391	220	234	150	6172	1	0	0	0	0	0	0	0	0	0	
6	2579	132	6	300	-15	67	230	237	140	6031	1	0	0	0	0	0	0	0	0	0	
7	2606	45	7	270	5	633	222	225	138	6256	1	0	0	0	0	0	0	0	0	0	
8	2605	49	4	234	7	573	222	230	144	6228	1	0	0	0	0	0	0	0	0	0	
9	2617	45	9	240	56	666	223	221	133	6244	1	0	0	0	0	0	0	0	0	0	
10	2612	59	10	247	11	636	228	219	124	6230	1	0	0	0	0	0	0	0	0	0	
11	2612	201	4	180	51	735	218	243	161	6222	1	0	0	0	0	0	0	0	0	0	

۱ تعدادی از داده‌ها

"Elevation" بلندی

"Aspect"

"Slope" شیب

"Horizontal_Distance_To_Hydrology" فاصله افقی تا نزدیک‌ترین سطح آب

"Vertical_Distance_To_Hydrology" فاصله عمودی تا نزدیک‌ترین سطح آب

"Horizontal_Distance_To_Roadways" فاصله افقی تا نزدیک‌ترین جاده

"Hillshade_9am"

"Hillshade_Noon"

"Hillshade_3pm"

"Horizontal_Distance_To_Fire_Points" فاصله افقی تا نزدیک‌ترین نقطه قابل اشتعال

"Wilderness_Area" که شامل ۴ دسته: Rawah, Neota, Comanche Peak, Cache la Poudre

"Soil_Type" که شامل ۴۰ نوع می‌باشد.

"Cover_Type" که شامل ۷ نوع می‌باشد:

- Spruce/Fir
- Lodgepole Pine
- Ponderosa Pine
- Cottonwood/Willow
- Aspen
- Douglas-fir
- Krummholz

که از ما خواسته شده است تا ستون "Cover_Type" را بدست آوریم.

۲. راه حل پیشنهادی

در ابتدا کلا داده‌ها را به شبکه عصبی دادم:

```
train.rf <- nnet(as.factor(Cover_Type)~. , data = train, size = 27,decay=.4, MaxNWts= 2000, maxit = 1000)

pre <- predict(train.rf, test, type = "class")
```

که ۶۰,۱۵۲٪ شد.

که "size" و "decay" (نرخ یادگیری) براساس پروژه قبلی دادم.

بعد از آن با استفاده از PCA خواستم داده‌ها را بهتر کنم و با همان شبکه عصبی:

```
transData <- preProcess(train[,1:54],  c("BoxCox", "center", "scale"))

predictorsTransData = data.frame(trans = predict(transData, train[,1:54]))

transTarget = preProcess(test,  c("BoxCox", "center", "scale"))

predictorsTransTarget = data.frame(trans = predict(transTarget, test))

train.rf <- nnet(as.factor(train$Cover_Type)~. , data = predictorsTransData, size = 27,decay=.1, MaxNWts= 2000, maxit=1000)

pre <- predict(train.rf, predictorsTransTarget, type = "class")
```

که ۳۸٪ شد.

یکبار دیگه هم "decay" را ۰,۱ دادم که انگار همگرا نمی‌شد.

با ctree و بدون PCA امتحان کردم:

```
train.rf <- ctree(as.factor(train$Cover_Type) ~. , data = train)

pre <- predict(train.rf,test, type = "response")
```

که ۶۰,۶۷٪ شد.

با randomForest امتحان کردم:

```
train.rf <- randomForest(as.factor(train$Cover_Type) ~. , data = train)
pre <- predict(train.rf,test, type = "response")
```

که ۷۰,۳۶٪ شد.

با svm امتحان کردم:

```
train.rf <- svm(as.factor(train$Cover_Type) ~. , data = train, type = "nu-classification")
pre <- predict(train.rf,test)
```

که ۴۹,۹۴٪ شد.

به فکر یکی کردن ستون‌ها یا حذف آن‌ها با استفاده از randomForest و تابع importance آن، ستون‌هایی که زیر ۱۰ بودن را حذف کردم، داده‌های جدید را دوباره به randomForest دادم که بهتر شد، ۷۲,۳۴٪ بدست آمد.

یکبار دیگر فقط بالای ۵۰ را نگه داشتم که ۷۲,۷۹٪ شد.

این‌بار به جای اینکه در دفعه دوم هم از randomForest استفاده کنم از svm استفاده کردم که به ۶۱,۶٪ رسیدم.

به همین ترتیب حال به جای svm از nnet استفاده کردم که ۵۶,۸٪ و با cforest به ۳۰,۲۴٪ رسیدم.

با استفاده از یک کد رسیدم به ۷۶٪ (درقسمت انتهایی آورده شده است).

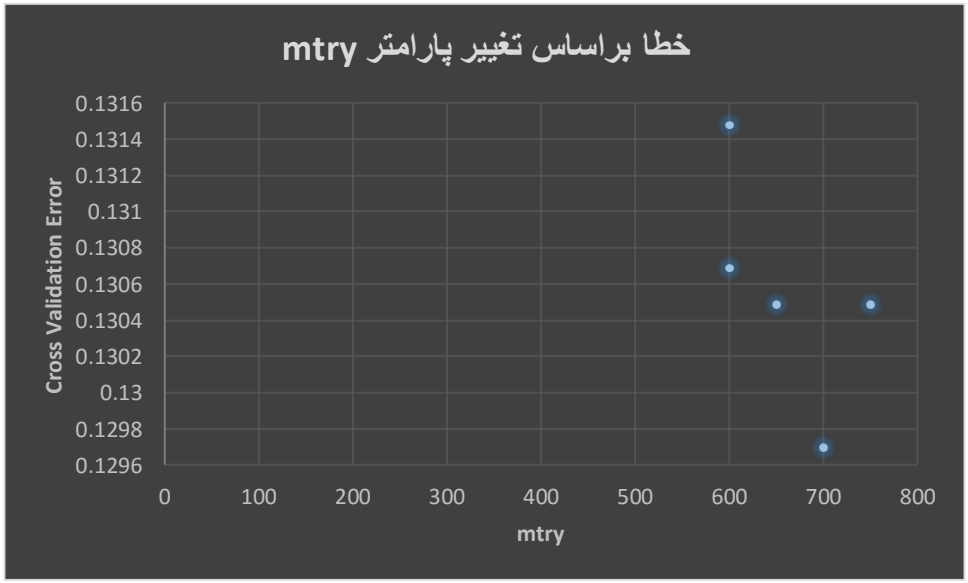
از چیزایی که در این کد با کار من فرق داشت:

- randomForest را با mtry = ۱۸ و ntree = ۶۰۰ اجرا کرده بود.
- یکسری از ستون‌هایی که correlation > ۶۰ را حذف کرده بود و یکی از آن‌ها را نگه داشته بود.(که شاید از نقاط قوت این کد به حساب می‌آمد)

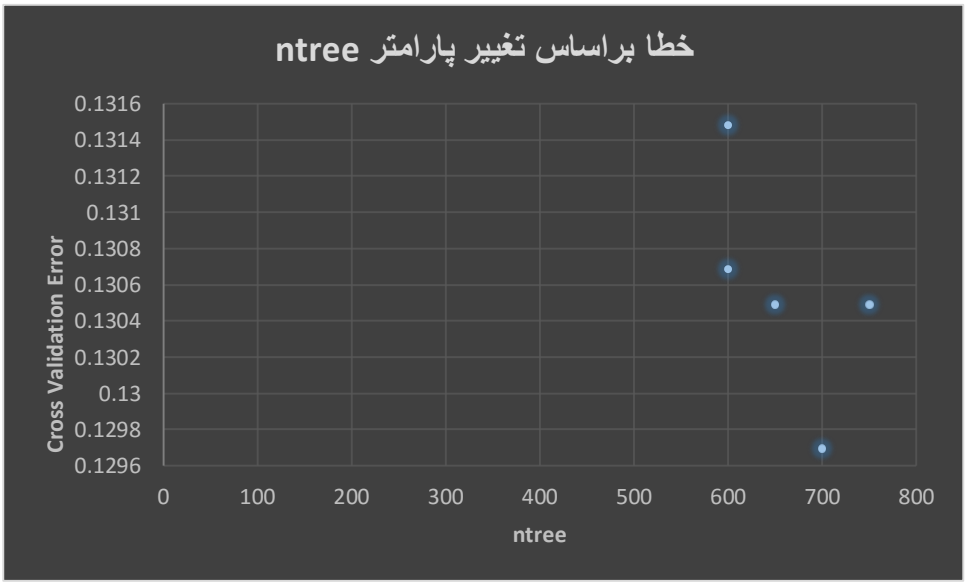
بعد از آن سعی کردم با استفاده از تابع tune.randomForest از کتابخانه e1071 mtry و ntree مناسب پیدا کنم برای بهبود دادن کد که در نهایت به ۷۵٫۵٪ رسیدم.

در ابتدا فقط با تغییر دادن mtry نتیجه را مشاهده کردم:

mtry	Cross validation error
۲۸	۰٫۱۳۵۵۱۵
۳۰	۰٫۱۲۹۴۳۱
۳۷	۰٫۱۳۱۹۴۴
۳۵	۰٫۱۳۱۱۵۰
۳۳	۰٫۱۲۸۹۰۲



ntree	Cross validation error
۶۰۰	۰,۱۳۰۶۸۷
۶۰۰	۰,۱۳۱۴۸۱
۶۵۰	۰,۱۳۰۴۸۹
۷۰۰	۰,۱۲۹۶۹۵
۷۵۰	۰,۱۳۰۴۸۹



۳. کد کامل

کد از منبع خارجی:

```
train <- read.csv("C:/Users/Hamed/Desktop/Data Science/Kaggle/Forest Cover Type Prediction/Question/train.csv")

test <- read.csv("C:/Users/Hamed/Desktop/Data Science/Kaggle/Forest Cover Type Prediction/Question/test.csv")

train1<-train[,(12:56)]

train1<-lapply(train1,factor)

train[,(12:56)]<-train1

# Remove column 22 as it has only one factor for all


train<-train[,-22]

#str(train)


# Remove column 29 as it has only one factor for all

train<-train[,-29]

#str(train)


# Remove column 1 id not of any use


train<-train[,-1]

#str(train)


fun<-function(x){
  x<-x/25
}
```

```
train[,7:9]<-sapply(train[,7:9],fun)
#pairs.panels(train[,1:10])


# this was to visualise the relationship between the vectors.
#finding corelation between one and another... very imp part as
#it help in eleminiating the multi collinearity in the dataframe and help in vector selection


library(randomForest)
# removed items with multicollinearity
mrf1<-randomForest(Cover_Type ~
Elevation+Aspect+Slope+Horizontal_Distance_To_Hydrology+Horizontal_Distance_To_Road
ways+Horizontal_Distance_To_Fire_Points+Wilderness_Area1+Wilderness_Area2+Wilderness
_Area3+Wilderness_Area4+Soil_Type1+Soil_Type2+Soil_Type3+Soil_Type4+Soil_Type5+Soi
l_Type6+Soil_Type8+Soil_Type9+Soil_Type10+Soil_Type11+Soil_Type12+Soil_Type13+Soil
_Type14+Soil_Type16+Soil_Type17+Soil_Type18+Soil_Type19+Soil_Type20+Soil_Type21+
Soil_Type22+Soil_Type23+Soil_Type24+Soil_Type25+Soil_Type26+Soil_Type27+Soil_Type
28+Soil_Type29+Soil_Type30+Soil_Type31+Soil_Type32+Soil_Type33+Soil_Type34+Soil_T
ype35+Soil_Type36+Soil_Type37+Soil_Type38+Soil_Type39+Soil_Type40, data = train,
ntree=700, mtry=33)


test1<-test[,(12:55)]
test1<-lapply(test1,factor)
test[,(12:55)]<-test1


# Remove column 22 as it has only one factor for all
test<-test[,-22]


# Remove column 29 as it has only one factor for all
test<-test[,-29]
test1<-test[,1]
```

```
# Remove column 1 id not of any use
test<-test[,-1]
test[,7:9]<-sapply(test[,7:9],fun)
predict<-predict(mrf1,newdata=test)
predict<-as.data.frame(predict)
test1<-as.data.frame(test1)
colnames(test1)<-c("id")
test1$Cover_Type<-predict$predict
write.csv(test1,file="C:/Users/Hamed/Desktop/Data Science/Kaggle/Forest Cover Type Prediction/Question/output.csv",row.names=F)
```

کد نوشته خودم:

```
library(nnet)
library(party)
library(randomForest)
library(e1071)
set.seed(65)
train <- read.csv("C:/Users/Hamed/Desktop/Data Science/Kaggle/Forest Cover Type Prediction/Question/train.csv")
test <- read.csv("C:/Users/Hamed/Desktop/Data Science/Kaggle/Forest Cover Type Prediction/Question/test.csv")

train <- data.frame(train)
train <- train[,-1]

test <- data.frame(test)
test <- test[,-1]
```

```
#with PCA

#transData <- preProcess(train[,1:54],  c("BoxCox", "center", "scale"))

#predictorsTransData = data.frame(trans = predict(transData, train[,1:54]))

#transTarget = preProcess(test,  c("BoxCox", "center", "scale"))

#predictorsTransTarget = data.frame(trans = predict(transTarget, test))

#train.rf <- nnet(as.factor(train$Cover_Type)~. , data = predictorsTransData, size = 27,decay=.1, MaxNWts= 2000, maxit=1000)

#pre <- predict(train.rf, predictorsTransTarget, type = "class")

#train.rf <- cforest(as.factor(Cover_Type) ~ ., data=train,  control = cforest_unbiased(ntree = 50))

#pre <- predict(train.rf, test, OOB=TRUE, type = "response")

#pre1 <- predict(train.rf, test[1:10000,], OOB=TRUE, type = "response") ba in khub kar kard!

#pre2 <- predict(train.rf, test[300001:565892,], OOB=TRUE, type = "response")

#train.rf <- nnet(as.factor(Cover_Type)~. , data = train, size = 27,decay=.01, MaxNWts= 2000, maxit=1000)

#pre <- predict(train.rf, test, type = "class")


#train.rf <- randomForest(as.factor(train$Cover_Type) ~. , data = train)

#pre <- predict(train.rf,test, type = "response")

train.rf <- randomForest(as.factor(train$Cover_Type) ~. , data = train, ntree=600, mtry=18)

pre <- predict(train.rf,test, type = "response")



high_importance <- which(importance(train.rf) > 50)

train1 <- cbind(train[,high_importance], train[55])

test1 <- cbind(test[,high_importance])
```

```
train1.rf <- randomForest(as.factor(train1$Cover_Type) ~. , data = train1, ntree=600, mtry=18)

pre1 <- predict(train1.rf,test1, type = "response")


#train.svm <- svm(as.factor(train1$Cover_Type)~. , data = train1, type = "nu-classification")

#pre.svm <- predict(train.svm, test1)


#train.nnet <- nnet(as.factor(Cover_Type)~. , data = train1, size = 15,decay=.01, MaxNWts=
2000, maxit=1000)

#pre.nnet <- predict(train.nnet, test1, type = "class")


#train.cf <- cforest(as.factor(Cover_Type) ~ ., data=train1,  control = cforest_unbiased(ntree =
50))

#pre.cf <- predict(train.cf, test1, OOB=TRUE, type = "response")


t <- 15121:581012

t <- data.frame(t)

t[,2] <- pre1

colnames(t) <- c("Id","Cover_Type")

write.csv(t, file = "C:/Users/Hamed/Desktop/Windows/Data Science/Kaggle/Forest Cover Type
Prediction/Question/submission.csv", row.names = FALSE)
```