

گزارش پروژه پوشش جنگلی



حامد زارعی

نام شرکت: فرابرد شبکه

نام مسئول: سید محمد غفاریان

پیش‌ بینی پوشش جنگلی

فهرست مطالب

[1. شرح مسئله 2](#_Toc427681462)

[2. راه حل پیشنهادی 4](#_Toc427681463)

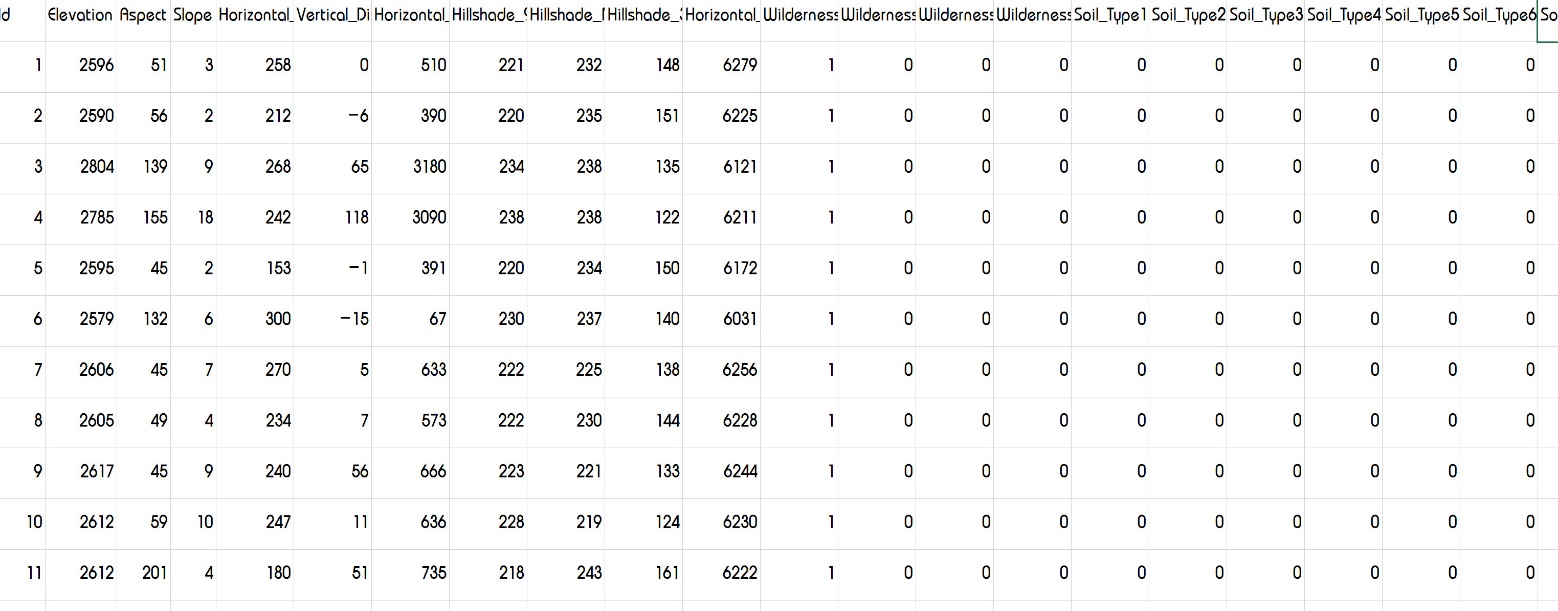
[3. کد کامل 8](#_Toc427681464)

# شرح مسئله

در این چالش از ما خواسته شده تا نوع پوشش جنگلی که در 7 دسته تقسیم بندی شده‌اند را پیش‌بینی کنیم.

داده‌های موجود در داده‌های آموزش از زیر نظر گرفتن یک بخش30 از 4 ناحیه صحرایی قرار گرفته در

جنگل Roosevelt National در شمال Colorado بدست آماده است.

 که تعدادی از داده‌ها در شکل زیر آورده شده است:

1 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 4 دسته: Rawah, Neota, Comanche Peak, Cache la Poudre

“Soil\_Type” که شامل 40 نوع می‌باشد.

“Cover\_Type” که شامل 7 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.

# 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 60.152% 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 38% شد.

یکبار دیگه هم “decay” را 0.1 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 60.67% شد.

با randomForest امتحان کردم:

train.rf <- randomForest(as.factor(train$Cover\_Type) ~. , data = train)

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

که 70.36% شد.

با svm امتحان کردم:

train.rf <- svm(as.factor(train$Cover\_Type) ~. , data = train, type = "nu-classification")

pre <- predict(train.rf,test)

که 49.94% شد.

به فکر یکی کردن ستون‌ها یا حذف آن‌ها با استفاده از randomForest و تابع importance آن، ستون‌هایی که زیر 10 بودن را حذف کردم، داده‌های جدید را دوباره به randomForest دادم که بهتر شد، 72.34% بدست آمد.

یکبار دیگر فقط بالای 50 را نگه داشتم که 72.79% شد.

این‌بار به جای اینکه در دفعه دوم هم از randomForest استفاده کنم از svm استفاده کردم که به 61.6% رسیدم.

به همین ترتیب حال به جای svm از nnet استفاده کردم که 56.8% و با cforest به 30.24% رسیدم.

با استفاده از یک کد رسیدم به 76% ( درقسمت انتهایی آورده شده است).

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

* randomForest را با mtry = 18 و ntree = 600 اجرا کرده بود.
* یکسری از ستون‌هایی که correlation < 60 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 75.5% رسیدم.

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

|  |  |
| --- | --- |
| Cross validation error | mtry |
| 0.135515 | 28 |
| 0.129431 | 30 |
| 0.131944 | 37 |
| 0.131150 | 35 |
| 0.128902 | 33 |

|  |  |
| --- | --- |
| ntree | Cross validation error |
| 600 | 0.130687 |
| 600 | 0.131481 |
| 650 | 0.130489 |
| 700 | 0.129695 |
| 750 | 0.130489 |

# 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\_Roadways+Horizontal\_Distance\_To\_Fire\_Points+Wilderness\_Area1+Wilderness\_Area2+Wilderness\_Area3+Wilderness\_Area4+Soil\_Type1+Soil\_Type2+Soil\_Type3+Soil\_Type4+Soil\_Type5+Soil\_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\_Type28+Soil\_Type29+Soil\_Type30+Soil\_Type31+Soil\_Type32+Soil\_Type33+Soil\_Type34+Soil\_Type35+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)