34

Let's learn R together! Kaggle's "Titanic" competition.

(self.MachineLearning)
submitted 10 months ago by [deleted]

Kaggle currently has a "knowledge" competition going where you are given some basic demographic information about passengers and are asked to use this information to predict whether or not they survived the disaster.

Here's a link to the competition: http://www.kaggle.com/c/titanic-gettingStarted

This is a great learning problem because the data set is relatively simple, with some missing values thrown in, and just enough noise to make things interesting.

I'm just learning R and here's what I propose: If you're interested in honing your R skills (or are just learning predictive modeling) we try to tackle this problem together. In this thread we will post our R models--with plenty of comments explaining what each line of code does and why we chose to do it. In addition, we'll post our scores so everyone can see how successful our models are. To start things off, I'll post a comment with my best scoring randomForest model.

Since this is a learning thread, if you don't understand something about any code that's posted, just ask! R has a slight learning curve and everyone has to start somewhere.

19 comments **share**

all 19 comments

sorted by: **best**

[–] **[deleted]** 10 points 10 months ago*

Score: 0.78469 Model: randomForest

The first thing I did was try to clean the data a bit. In both the "train" data and the "test" there are a number of 'NA' fields (fields missing data). I decided to disregard things like passenger name

MachineLearning

subscribe 23,947 readers

~28 users here now

News, Research Papers, Videos, Lectures, Softwares and Discussions on:

- Machine Learning
- Data Mining
- Information Retrieval
- Predictive Statistics
- Learning Theory
- Search Engines
- Pattern Recognition
- Analytics

Beginners:

Please have a look at our FAQ and Link-Collection

Related Subreddit :

- Statistics
- Computer Vision
- Compressive Sensing
- MLClass
- NLP

and fare. Passenger name because it's very noisy, fare because it's missing a lot of data and looks like it means "balance paid" rather than individual fare. So for a family of four, if one person paid for everything, their fare might look really high even for 3rd class passage. So the two columns with missing data are "age" and "embarked."

I looked for the median age of the passengers and used this median to fill in any NA ages. I then looked for the most common embarkation code and used that to fill in any NA embarked fields.

After the massaging, here's the code I used.

```
> library(randomForest)

#now I load in both the "train" and "test" data f

> train <- read.csv("C:/train.csv")
> test <- read.csv("C:/test.csv")

#Now, randomForest can either model a regression
#This competition is to predict survival. Either
#So, I want to make sure randomForest models a cl
#that the "Survived" column (our dependent variab

> train$Survived <- as.factor(train$Survived)

#Next, I decide what predictors (independant vari
#I decided to use Pclass, Sex, Age, SibSp, Parch,

#Additionally, I decide to include the relationsh
#just for kicks. I think women are more likely to
#but I think it's possible that a rich woman (Pcl
#poor woman. So we'll model things like this in a

#Before I do anything else, I want to set a "seed
#randomForest is RANDOM. If I don't set a seed va
#get a slightly different result.

> set.seed(107)

#Now I build my model

> model <- randomForest(Survived ~ Pclass + Sex +

#Now I use the model to predict against test data
#called “Survived” in the test data frame.

> test$Survived <- predict(model, newdata=test, t

#Finally I save my updated test data frame as a .

> write.csv(test, file=”c:/R/predictions.csv”, ro

#Now, outside of R I go into my newly saved "pred
#except for passenger ID and Survived. I then upl
```

Score for this model: 0.78469

permalink

[–] **BruceJillis** 5 points 10 months ago*

```
write.csv(test, file=”c:/R/predictions.csv”, row.names=F)
```

- Jobs

created by kunjaan a community for 4 years

MODERATORS message the moderators

kunjaan
kanak
cavedave
olaf_nij
BeatLeJuce

Hey great idea, thanks for submitting your code as well! I have participated before in kaggle but always using python but I have been looking for an excuse to get into R. Quick tip, this line will output the csv in one fell swoop ready for submission:

```
write.csv(test[,c("PassengerId", "Survived")], file="predictions.csv", row.names=FALSE, (
```

permalink **parent**

[–] **[deleted]** 1 point 10 months ago

Fantastic, thanks!

permalink **parent**

[–] **madamfunk** 3 points 10 months ago

Thanks for sharing grendelgrey, I did something similar a couple months ago and scored .79 , the only difference is I did more data imputation.

permalink **parent**

[–] **[deleted]** 1 point 10 months ago

I'd really like to learn to use the rfImpute function in the randomForest package. Any chance you could post some R code about how to do this?

permalink **parent**

[–] **[deleted]** 1 point 10 months ago

I think this model could perform better with a more sophisticated NA replacement scheme. I plan to try the following tomorrow:

For finding missing ages I originally just used the median age of all passengers. However, the data set gives us more to work with than that. The Name column contains honorifics, which can be used to make better age estimates. I want to break down the ages by honorific (Mr. for a man, Mrs. for a woman, Miss for a young woman, Master for a young man, etc.)

I'll then take the median age for each honorific category and use that to fill in the missing ages.

I'm not really sure if there's a better way to fill in missing embarkation data.

I'd LOVE to figure out a way to use Cabin but there are so many missing data that 'm not sure how. If anyone figures out a way, please let us know!

permalink **parent**

[–] **unoogle** 1 point 7 months ago

I tried your code but I get:

> test$Survived <- predict(model, newdata=test, type='class') Error: unexpected input in "test$Survived <- predict(model, newdata=test, type='"
> test$Survived=predict(model,newdata=test) Error in predict.randomForest(model, newdata = test) : Type of predictors in new data do not match that of the training data.

permalink **parent**

[–] **[deleted]** 4 points 10 months ago

Oh, I also just want to tell people not to worry too much about where on the leader board you land. There are scores on there of .9+ ... Ignore them. Those people likely cheated by looking up survivor names on the Internet or just over fitting their models by submitting over and over again. I think for this competition a score of around .80-.86 is likely the best you could possibly do.

permalink

[–] **[deleted]** 1 point 9 months ago

Yeah I saw phd students in that competition using rather robust methods and still getting 79-86. I've maxed out at 79, after trying a sloppy logit model and then running imputed test and training data through a random forest model. You may be right about people over fitting or just being dicks

permalink **parent**

[–] **BruceJillis** 3 points 10 months ago*

Ok, so it's getting late over here so i'm calling it a day. I'm definitely going to be doing some more work on this, I had fun. I found this to be a very useful page to have open. My code is here. Right now there is:

- a slightly modified version of the randomforests script (best score: 0.78947)
- a logistic regression model (best score 0.77990)
- a data imputation script that does simple random data imputation and topcoding.
- tried out some imputation packages (could only get VIM, rrcovna and mice to work, included some small examples) but settled on Hmisc's impute function for now
- started working through this logistic regression tutorial

permalink

[–] **[deleted]** 1 point 10 months ago

Great! Thanks for the links!

permalink **parent**

[–] **[deleted]** 2 points 10 months ago

Great idea for a thread. I'll give it a try soon and post the results here.

permalink

[–] **Fogrocket** 2 points 10 months ago

I love this thread already! I am going to observe, copy/recreate and learn. Then I'll try a model later. Note: I've used R a little by force from my masters degree but I'm a minitab user so this is going to be a nice exercise for me! Thanks OP

permalink

[–] **[deleted]** 1 point 10 months ago

Cool, good luck! This competition has taught me a lot about R already. I'm looking forward to seeing what we can come up with!

permalink **parent**

[–] **denacioust** 2 points 10 months ago

I used the RWeka package and the J4.8 machine learning algorithm. I only came across this recently and I have no idea of what it does exactly, I just know it gives reasonably good results. It got a score of 0.76 on Kaggle. This might not give anyone any help with statistical knowledge but might help people learn some bits of R.

```
install.packages("RWeka")
library(RWeka)
#Setting factor variables as factors using loops
fs <- c(2,3,5,7,8,9,10,11,12)
for(i in 1:length(fs)){
train[,i] <- as.factor(train[,i])
}
#Removing any rows with NAs
train <- train[complete.cases(train),]
#Fitting the model
model <- J48(Survived~.,data=train)
#Making predictions
test$Survived <- predict(model, newdata=test,type="class")
#Saving predictions
write.csv(test[,c("PassengerId", "Survived")], file="predictions.csv", row.names=FALSE, quot
```

permalink

[–] **[deleted]** 1 point 10 months ago

Neat. I hadn't heard of RWeka before. I'll definitely check it out.

permalink **parent**

[–] **manueslapera** 1 point 10 months ago

man, I have been trying to install weka for 3 months in my ubuntu. Damn you Rjava!

permalink **parent**

[–] **srepho** 2 points 9 months ago*

I am only quite new to kaggle but here is some useful things that I found in the titanic competition. Firstly I highly recommend the caret package in R. It is brilliant and really streamlines the model building process.

-Random Forest.
library(caret)
library(randomForest)

I will skip the loading of data and converting to factors and dropping of name, ticket and cabin.

In random forests there are two tuning parameters mtry which is the number of predictors that are used for each tree and ntree which is the number bootstrap samples. The great thing about Random Forests is I don't believe it is possible to overfit using too many ntrees (you just reach a plateau of performance). Lets ask Caret to fine tune the mtry for us.
rfmodel<-train(train$survived~., data=train, method="rf")

then you can just look at the results using

rfmodel and summary(rfmodel)

A couple of other things that are important (and that caret does really well) are:

-Resampling not only can improve most models, it also gives you a better idea of what you score will be on the test data. Cross validation in its various forms and bootstraping and other similair techniques are really important to include in model building. In caret you set this within the code itself as I will show with a SVM model (which I think is my highest scoring model in this competition ~98th though this need extra imputs created by working on the initial data set).

install.packages("kernlab")
install.packages("caret")
library(kernlab)
library(caret)

svmmodel<-train(train$survived~., data=train,
+ method="svmRadial",
+ metric="Accuracy",
+ preProc=c("knnImpute", "center", "scale")
+ trControl=trainControl(method="repeatedcv", repeats=5)
+ tuneLength = 10)

titanicpred<-predict(svmmodel, newdata=test)

-I found in this competition that there comes a point where you cannot do any better without including some extra data from what is discarded. If you read through the forums you will some people have kindly shared some discoveries they have made - in particular spoiler is useful.

-Have a look at ensembling across models. Most competitions are won by people who put together many different models and then use their input as one kind of meta-model. A simple way of doing this is to simply take the output of some models and just take the majority vote. If most models think a person would have died then mark it as dead etc. There are many ways to put the models together so it is worth playing around with. As far as I am aware there are no packages in R that do it automatically - but keep an eye on caretEnsemble (not written by the person who wrote caret) which is in development.

-I also highly recommend the book Applied Predictive Modeling (one of the co-authors is the creator of caret). It has made a massive difference both to my kaggling and my model building. Best book for kagglers I have seen.

Anyway hope at least some of the above helps - best of luck with the competition!

Stephen

permalink

[–] **BruceJillis** 2 points 9 months ago

Caret's confusionMatrix is also very nice! I saw it being used somewhere and have been using it ever since but haven't checked out the rest.. I see I have been missing out. Thanks for the writeup, i'm definitely going to check out some of your suggestions.

permalink **parent**