

IMT 573: Problem Set 8

Prediction

Jenny Skytta

Due: May 22, 2022

Collaborators:

Instructions: Before beginning this assignment, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Cloud.

1. Download the `08_ps_prediction.Rmd` file from Canvas or save a copy to your local directory on RStudio Cloud. Supply your solutions to the assignment by editing `08_ps_prediction.Rmd`.
2. Replace the “YOUR NAME HERE” text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chunks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do not need four different visualizations of the same pattern.
4. Collaboration on problem sets is fun and useful, and we encourage it, but each student must turn in an individual write-up in their own words as well as code/work that is their own. Regardless of whether you work with others, what you turn in must be your own work; this includes code and interpretation of results. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students’ responses or code.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run without errors you can do so with the `eval=FALSE` option as follows:

```
a + b # these object don't exist
# if you run this on its own it will give an error
```

7. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click Knit, download and rename the knitted PDF file to `ps7_YourLastName_YourFirstName.pdf`, and submit the PDF file on Canvas.

Setup: In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(tidyverse)
library(nycflights13) # loading nycflights13 library
```

```
library(gridExtra)
library(MASS)
library(pROC)
library(arm)
library(randomForest)
library(Metrics)
library(knitr) # this will keep code on the page!
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Data: In this problem set we will use the `flights` dataset used previously in class. The `flights` dataset (via the `nycflights13` library) contains information on flight delays and weather.

As part of this assignment, we will evaluate the performance of several statistical learning methods. We will fit our learning models using a set of *training* observations and measure its performance on a set of *test* observations.

Problem 1: Discuss the advantages of using a training/test split when evaluating statistical models.

From our module reading, some advantages of using a training / test split when evaluating statistical models might be that it could help us avoid overfitting as well as improve accuracy of our training data. Before we utilize any real data, having that internal validity testing ensures that the models can work.

Problem 2: Predictions with a continuous output variable

```
data(flights) # loading flights dataset
data(weather) # loading weather dataset

# using an inner join to combine the datasets on the specific variables
flight_weather_join <- inner_join(flights, weather, by = c("origin", "year", "month", "day", "hour"))
```

(a) Load in the `flights` dataset. Join the `flights` data to the `weather` data based on the departure location, date, and hour of the flight. Exclude data entries which cannot be joined to `weather` data. Copy the joined data so we can refer to it later.

```
# subsetting the dataframe into smaller tibble with specified column variables
flight_weather_select = subset(flight_weather_join, select = c(dep_delay, origin, dep_time, temp, wind_
  na.omit())
```

(b) From the joined data, keep only the following columns as we build our first model: departure delay, origin, departure time, temperature, wind speed, precipitation, and visibility. Omit observations that do not have all of these variables present.

```
# pulling out n for 80% of observations
split <- round(nrow(flight_weather_select) *0.80)
# setting seed with that value
set.seed(261518)
```

```
# generating random sample
rows <- sample(1:nrow(flight_weather_select), 261518)
# allocating the training set
Training <- flight_weather_select[rows, ]
# allocating the test set
Test <- flight_weather_select[(split + 1):nrow(flight_weather_select), ]
```

(c) Split your data into a *training* and *test* set based on an 80-20 split. In other words, 80% of the observations will be in the training set and 20% will be in the test set. Remember to set the random seed.

```
# creating a linear model of the training subset with y of dep_delay
mtraining <- lm(dep_delay ~ origin + dep_time + temp + wind_speed + precip + visib, data = Training)
summary(mtraining)
```

(d) Build a linear regression model to predict departure delay using the subset of variables indicated in (3.). What is the RMSE on the training set? What is the RMSE on the test set? Which is higher and is this expected?

```
##
## Call:
## lm(formula = dep_delay ~ origin + dep_time + temp + wind_speed +
##      precip + visib, data = Training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -114.22  -18.81   -8.17    2.82  1305.93
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.1336562  0.4894271  -8.446  <2e-16 ***
## originJFK   -4.5777337  0.1834630 -24.952  <2e-16 ***
## originLGA   -4.1971291  0.1852808 -22.653  <2e-16 ***
## dep_time     0.0212572  0.0001564 135.902  <2e-16 ***
## temp         0.1166783  0.0042850  27.229  <2e-16 ***
## wind_speed   0.2688598  0.0140971  19.072  <2e-16 ***
## precip       71.0462307  2.6776351  26.533  <2e-16 ***
## visib       -2.0412392  0.0405198 -50.376  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.4 on 261510 degrees of freedom
## Multiple R-squared:  0.08857,    Adjusted R-squared:  0.08854
## F-statistic: 3630 on 7 and 261510 DF,  p-value: < 2.2e-16
```

```
# RMSE of training dataset
RMSE_train <- sqrt(mean((Training$dep_delay - predict(mtraining, Training)) ^ 2))
# RMSE of test dataset
RMSE_test <- sqrt(mean((Test$dep_delay - predict(mtraining, Test)) ^ 2))
# RMSE difference in absolute value
RMSE_diff <- abs(RMSE_test - RMSE_train)
```

The RMSE for training was 38.4008791 while the RMSE for test was 38.0562236. They are nearly the same value but the test is larger by 0.3446555.

```
# subsetting the dataframe into smaller tibble with specified column variables
flight_weather_more = subset(flight_weather_join, select = c(dep_delay, origin, dep_time, temp, wind_sp
  na.omit()
cor(flight_weather_more[, 16]))
```

(e) Now, improve upon these prediction results by including additional variables in your model. Make sure you keep at least 95% of original data (i.e. about 320K observations across both the training and test datasets). Do not include the arrival time, scheduled arrival time, or the arrival delay in your model. Use the same observations as above for the training and test sets (i.e. keep the same rows but add different variables/columns at your discretion). Can you improve upon the training RMSE? Once you have a model that you feel adequately improves the training RMSE, does your model improve the test RMSE? Which variables did you include in your model?

```
##          sched_dep_time
## sched_dep_time          1

# pulling out n for 80% of observations
split_all <- round(nrow(flight_weather_more) *0.80)
# setting seed with that value
set.seed(254204)
# generating random sample
rows2 <- sample(1:nrow(flight_weather_more), 254204)
# allocating the training set
Training_all <- flight_weather_more[rows2, ]
# allocating the test set
Test_all <- flight_weather_more[(split_all + 1):nrow(flight_weather_more), ]

# creating a linear model of the training subset with y of dep_delay
mtraining_all <- lm(dep_delay ~ origin + dep_time + temp + visib + dewp + air_time + carrier + dest + s
summary(mtraining_all)

##
## Call:
## lm(formula = dep_delay ~ origin + dep_time + temp + visib + dewp +
##      air_time + carrier + dest + sched_dep_time, data = Training_all)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -94.27  -16.50   -7.67    2.02  1308.92
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.204e+01  3.292e+00  -3.658 0.000254 ***
## originJFK    -1.078e+00  2.764e-01  -3.899 9.67e-05 ***
## originLGA    -5.126e-01  2.438e-01  -2.103 0.035503 *
## dep_time      6.448e-02  5.272e-04  122.314 < 2e-16 ***
## temp        -4.131e-01  1.042e-02  -39.647 < 2e-16 ***
## visib       -1.090e+00  4.288e-02  -25.410 < 2e-16 ***
## dewp         5.296e-01  9.528e-03   55.580 < 2e-16 ***
```

## air_time	-8.398e-03	6.809e-03	-1.233	0.217432	
## carrierAA	-5.329e+00	5.095e-01	-10.459	< 2e-16	***
## carrierAS	-7.397e+00	1.844e+00	-4.012	6.01e-05	***
## carrierB6	-8.519e-01	4.391e-01	-1.940	0.052341	.
## carrierDL	-5.417e+00	4.537e-01	-11.941	< 2e-16	***
## carrierEV	5.364e+00	4.557e-01	11.769	< 2e-16	***
## carrierF9	2.207e+00	1.788e+00	1.234	0.217064	
## carrierFL	1.761e+00	1.037e+00	1.697	0.089718	.
## carrierHA	-4.674e+00	3.284e+00	-1.423	0.154691	
## carrierMQ	-4.164e+00	4.755e-01	-8.758	< 2e-16	***
## carrierOO	-7.524e+00	8.419e+00	-0.894	0.371482	
## carrierUA	-3.031e+00	5.028e-01	-6.029	1.66e-09	***
## carrierUS	-7.250e+00	5.676e-01	-12.773	< 2e-16	***
## carrierVX	-1.444e-01	7.802e-01	-0.185	0.853160	
## carrierWN	4.562e+00	7.037e-01	6.483	9.00e-11	***
## carrierYV	-1.221e+00	1.916e+00	-0.638	0.523741	
## destACK	1.038e+01	3.969e+00	2.614	0.008940	**
## destALB	1.158e+01	3.686e+00	3.141	0.001682	**
## destANC	1.229e+01	1.449e+01	0.848	0.396532	
## destATL	1.618e+01	2.850e+00	5.677	1.38e-08	***
## destAUS	1.105e+01	2.810e+00	3.932	8.42e-05	***
## destAVL	4.037e+00	3.900e+00	1.035	0.300517	
## destBDL	8.470e+00	3.762e+00	2.251	0.024363	*
## destBGR	7.825e+00	3.745e+00	2.090	0.036651	*
## destBHM	9.531e+00	3.840e+00	2.482	0.013065	*
## destBNA	1.311e+01	2.881e+00	4.550	5.36e-06	***
## destBOS	1.057e+01	3.042e+00	3.475	0.000511	***
## destBQN	2.350e+01	3.037e+00	7.737	1.02e-14	***
## destBTW	1.057e+01	3.117e+00	3.391	0.000697	***
## destBUF	1.117e+01	3.034e+00	3.681	0.000232	***
## destBUR	7.945e+00	3.486e+00	2.279	0.022639	*
## destBWI	9.066e+00	3.209e+00	2.825	0.004728	**
## destBZN	1.823e+01	7.585e+00	2.404	0.016234	*
## destCAE	1.250e+01	5.281e+00	2.368	0.017888	*
## destCAK	1.435e+01	3.433e+00	4.180	2.91e-05	***
## destCHO	-9.749e-01	6.532e+00	-0.149	0.881353	
## destCHS	1.077e+01	2.980e+00	3.613	0.000303	***
## destCLE	1.216e+01	3.007e+00	4.045	5.24e-05	***
## destCLT	1.371e+01	2.923e+00	4.691	2.71e-06	***
## destCMH	1.123e+01	3.019e+00	3.720	0.000199	***
## destCRW	7.206e+00	4.678e+00	1.541	0.123439	
## destCVG	1.314e+01	2.955e+00	4.445	8.81e-06	***
## destDAY	1.117e+01	3.115e+00	3.584	0.000338	***
## destDCA	1.238e+01	3.047e+00	4.064	4.82e-05	***
## destDEN	1.632e+01	2.723e+00	5.991	2.08e-09	***
## destDFW	1.464e+01	2.741e+00	5.342	9.19e-08	***
## destDSM	1.079e+01	3.339e+00	3.232	0.001227	**
## destDTW	1.366e+01	2.931e+00	4.661	3.15e-06	***
## destEGE	2.085e+01	3.946e+00	5.283	1.27e-07	***
## destEYW	1.647e+01	9.779e+00	1.684	0.092090	.
## destFLL	1.547e+01	2.767e+00	5.592	2.25e-08	***
## destGRR	1.106e+01	3.280e+00	3.371	0.000750	***
## destGSO	1.073e+01	3.136e+00	3.422	0.000621	***
## destGSP	9.838e+00	3.271e+00	3.008	0.002631	**

## destHDN	2.941e+01	1.164e+01	2.528	0.011488	*
## destHNL	2.242e+01	4.280e+00	5.238	1.62e-07	***
## destHOU	1.266e+01	2.860e+00	4.426	9.59e-06	***
## destIAD	1.158e+01	3.063e+00	3.780	0.000157	***
## destIAH	1.433e+01	2.741e+00	5.228	1.71e-07	***
## destILM	1.607e+00	5.044e+00	0.319	0.750047	
## destIND	1.141e+01	3.011e+00	3.790	0.000151	***
## destJAC	4.040e+01	9.247e+00	4.369	1.25e-05	***
## destJAX	1.133e+01	2.923e+00	3.878	0.000105	***
## destLAS	1.309e+01	2.738e+00	4.782	1.73e-06	***
## destLAX	1.310e+01	2.736e+00	4.787	1.70e-06	***
## destLEX	-1.442e+01	3.767e+01	-0.383	0.701793	
## destLGB	9.147e+00	3.179e+00	2.877	0.004016	**
## destMCI	1.240e+01	2.913e+00	4.255	2.09e-05	***
## destMCO	1.344e+01	2.794e+00	4.811	1.50e-06	***
## destMDW	1.240e+01	2.961e+00	4.188	2.81e-05	***
## destMEM	1.240e+01	2.958e+00	4.191	2.78e-05	***
## destMHT	1.158e+01	3.348e+00	3.458	0.000544	***
## destMIA	1.510e+01	2.781e+00	5.432	5.58e-08	***
## destMKE	1.317e+01	2.954e+00	4.459	8.25e-06	***
## destMSN	1.516e+01	3.354e+00	4.519	6.21e-06	***
## destMSP	1.500e+01	2.801e+00	5.357	8.49e-08	***
## destMSY	1.260e+01	2.805e+00	4.491	7.10e-06	***
## destMTJ	2.901e+01	1.164e+01	2.493	0.012660	*
## destMVY	3.886e+00	4.258e+00	0.913	0.361461	
## destMYR	1.516e+01	6.157e+00	2.462	0.013806	*
## destOAK	6.501e+00	3.619e+00	1.796	0.072436	.
## destOKC	9.265e+00	3.605e+00	2.570	0.010172	*
## destOMA	1.208e+01	3.128e+00	3.861	0.000113	***
## destORD	1.684e+01	2.838e+00	5.932	3.00e-09	***
## destORF	1.058e+01	3.204e+00	3.302	0.000959	***
## destPBI	1.536e+01	2.803e+00	5.479	4.28e-08	***
## destPDX	1.270e+01	2.950e+00	4.306	1.66e-05	***
## destPHL	1.444e+01	3.245e+00	4.451	8.56e-06	***
## destPHX	1.474e+01	2.757e+00	5.346	8.99e-08	***
## destPIT	1.132e+01	3.073e+00	3.683	0.000231	***
## destPSE	5.156e+01	3.561e+00	14.479	< 2e-16	***
## destPSP	7.183e+00	1.009e+01	0.712	0.476533	
## destPVD	1.135e+01	3.816e+00	2.973	0.002948	**
## destPWM	1.123e+01	3.122e+00	3.597	0.000322	***
## destRDU	1.261e+01	2.978e+00	4.235	2.29e-05	***
## destRIC	1.348e+01	3.118e+00	4.324	1.54e-05	***
## destROC	1.201e+01	3.115e+00	3.855	0.000116	***
## destRSW	1.288e+01	2.823e+00	4.561	5.10e-06	***
## destSAN	1.313e+01	2.828e+00	4.643	3.43e-06	***
## destSAT	1.462e+01	3.137e+00	4.661	3.16e-06	***
## destSAV	1.143e+01	3.263e+00	3.503	0.000461	***
## destSBN	6.869e+00	1.560e+01	0.440	0.659681	
## destSDF	1.039e+01	3.120e+00	3.331	0.000864	***
## destSEA	1.340e+01	2.816e+00	4.758	1.95e-06	***
## destSFO	1.687e+01	2.768e+00	6.095	1.10e-09	***
## destSJC	2.270e+00	3.562e+00	0.637	0.524001	
## destSJU	1.683e+01	2.738e+00	6.146	7.97e-10	***
## destSLC	1.197e+01	2.804e+00	4.271	1.95e-05	***

```
## destSMF          7.150e+00  3.714e+00  1.925 0.054203 .
## destSNA          9.927e+00  3.111e+00  3.191 0.001420 **
## destSRQ          1.338e+01  3.002e+00  4.458 8.28e-06 ***
## destSTL          1.252e+01  2.867e+00  4.367 1.26e-05 ***
## destSTT          1.876e+01  3.268e+00  5.741 9.40e-09 ***
## destSYR          1.250e+01  3.179e+00  3.931 8.46e-05 ***
## destTPA          1.447e+01  2.797e+00  5.173 2.31e-07 ***
## destTUL          1.338e+01  3.675e+00  3.640 0.000272 ***
## destTVC          1.500e+01  5.380e+00  2.787 0.005314 **
## destTYS          1.141e+01  3.376e+00  3.379 0.000728 ***
## destXNA          1.010e+01  3.050e+00  3.311 0.000931 ***
## sched_dep_time -4.537e-02  5.547e-04 -81.785 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.56 on 254077 degrees of freedom
## Multiple R-squared:  0.1283, Adjusted R-squared:  0.1279
## F-statistic: 296.8 on 126 and 254077 DF,  p-value: < 2.2e-16

# RMSE of training dataset
RMSE_train_all <- sqrt(mean((Training_all$dep_delay - predict(mtraining_all, Training_all)) ^ 2))
# RMSE of test dataset
RMSE_test_all  <- sqrt(mean((Test_all$dep_delay - predict(mtraining_all, Test_all)) ^ 2))
# RMSE difference in absolute value
RMSE_diff_all <- abs(RMSE_test_all - RMSE_train_all)
```

Initially, I used a variable windgust that ended up being filled with NA missing values and reduced the entire set. I ran a coefficient correlation on the dataset to see which variables were significant. I removed wind speed, precipitation as it didn't appear to add predictive power and their removal seemed to lower the RMSE. The variable that did seem to make an impact was carrier. The new training RMSE is 37.5461728 and 36.7156133 with a difference of 0.8305595 which between the previous is a lower overall value.

EXTRA CREDIT: Predictions with a categorical output (classification)

In this problem our goal is to predict the survival of passengers. First, let's train a logistic regression model for survival that controls for the socioeconomic status of the passenger.

```
# loading data
titanic <- as_tibble(Titanic)
# pulling out n for 80% of observations
titanicsplit <- round(nrow(titanic) *0.80)
# setting seed with that value
set.seed(26)
# generating random sample
titan_rows <- sample(1:nrow(titanic), 26)
# allocating the training set
titan_training <- titanic[rows, ]
# allocating the test set
titan_test <- titanic[(split + 1):nrow(titanic), ]
```

(a) Load in the titanic data. Split your data into a *training* and *test* set based on an 80-20 split. In other words, 80% of the observations will be in the training set and 20% will be in the test set. Remember to set the random seed.

```

# creating a linear model of the training subset with y of dep_delay
#titan_model <- lm(Survived ~ Class, data = titan_training)
# summary(titan_model)
# # RMSE of training dataset
# titan_RMSE_train <- sqrt(mean((Training$survived - predict(mtraining, Training)) ^ 2))
# # RMSE of test dataset
# titan_RMSE_test <- sqrt(mean((Test$dep_delay - predict(mtraining, Test)) ^ 2))
# # RMSE difference in absolute value
# titan_RMSE_diff <- abs(RMSE_test - RMSE_train)

```

(b) Fit the model described above (i.e. one that only takes into account socioeconomic status) using the `glm` function in R.

(c) What might you conclude based on this model about the probability of survival for lower class passengers? I'm going to conclude that class did have an impact on survival overall as the historically we know they were the ones evacuated first.

(d) Predict the survival of passengers for each observation in your test set using the model fit in Problem 2. Save these predictions as `yhat`.

(e) Use a threshold of 0.5 to classify predictions. What is the number of false positives on the test data? Interpret this in your own words.

(f) Using the `roc` function, plot the ROC curve for this model. Discuss what you find.

(g) Suppose we use the data to construct a new predictor variable based on a passenger's listed title (i.e. Mr., Mrs., Miss., Master). Why might this be an interesting variable to help predict passenger survival?

(h) Fit a second logistic regression model including this new feature. Use the `summary` function to look at the model. Did this new feature improve the model?

(i) Comment on the overall fit of this model. For example, you might consider exploring when misclassification occurs.