Bob R. Gardner - R programming Class - 12/17/19

Executive Summary

In reviewing over 2,000 data points of graduating high school students, several models were evaluated to determine which model would predict the best **graduation rates.** Among four models evaluated below, the race model was determined to be the best predictor.

Model Selection

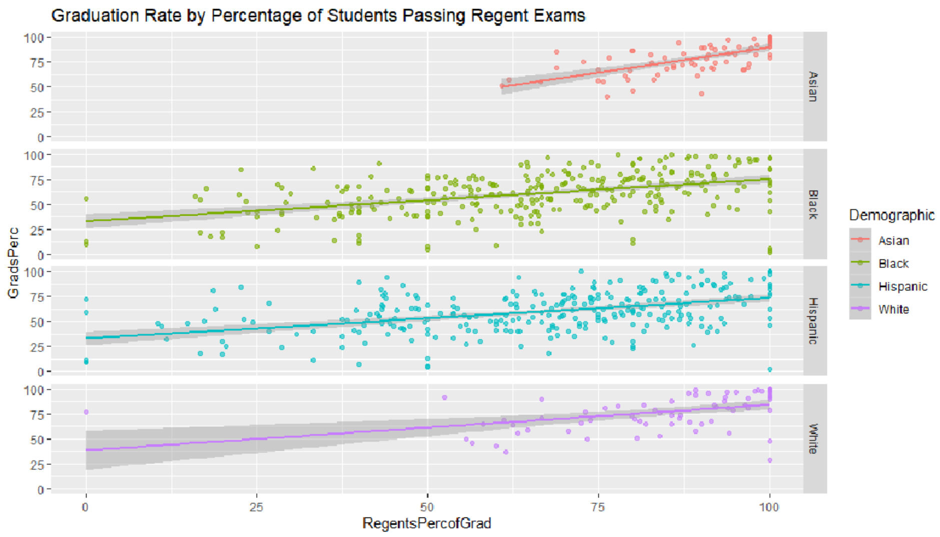
Several versions of models were tested to find the best choice. Initially several other variables (covariates) such as Borough designation (five boroughs within New York City) and School district (geographical regions within the city) were used. However, these were non-numeric inputs and the quality of the models (MSE) tend to the same even when more independent variables (explanatory) were added. Furthermore, the dummy variable (of a constant value of 1) had very similar values to the non-dummy variables. In other words, the control group and the experimental group were essentially the same.

The next iterations of models are seen below with summary statistics. Each model determined by 4 main criteria: race, student type, gender and English proficiency.

By Race

Independent Variable	Dependent Variable(s)	Fit (Adjusted R^2)	Mean Square Errors
Graduation Percentage	Percentage of Cohort Passing Regents	0.2594	420.439
	Drop Out Percentage	0.4341	266.1971
	Percentage of Cohort Passing Regents & Drop Out Percentage	0.5156	222.729
	Dummy Variable		4494.713

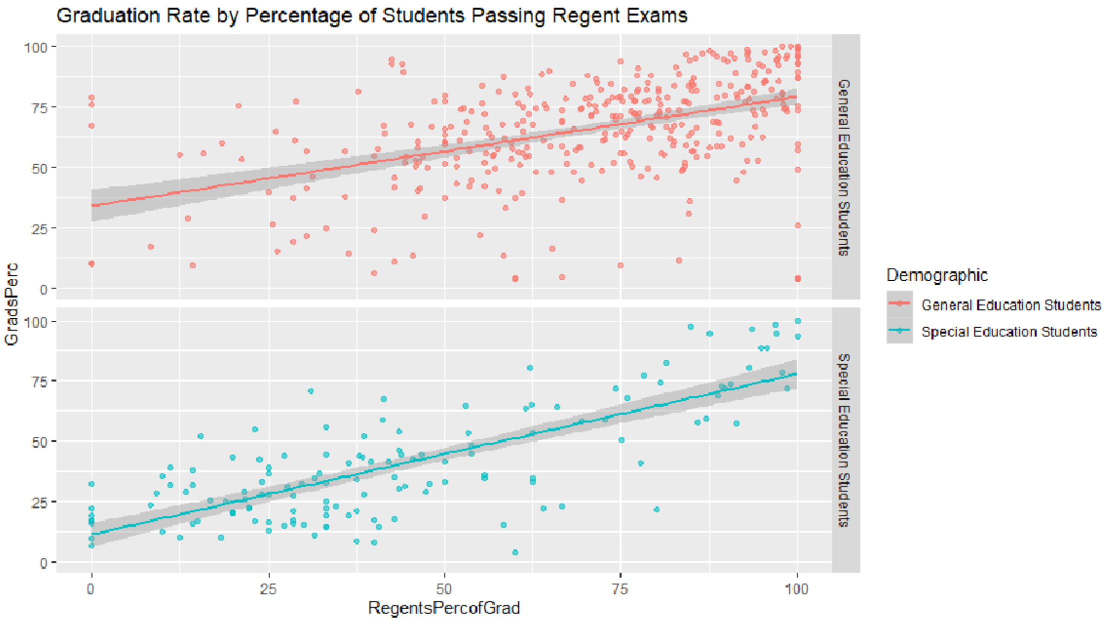
Within the race category of the Demographic field, the values were: Black, White, Hispanic and Asian. Not every school (understandably) had each ethnicity represented in their respective schools.



By Student Type

Independent Variable	Dependent Variable(s)	Fit (Adjusted R^2)	Mean Square Errors
Graduation Percentage	Percentage of Cohort Passing Regents	0.43	504.9257
	Drop Out Percentage	0.506	314.4553
	Percentage of Cohort Passing Regents & Drop Out Percentage	0.6001	249.5713
	Dummy Variable		4072.44

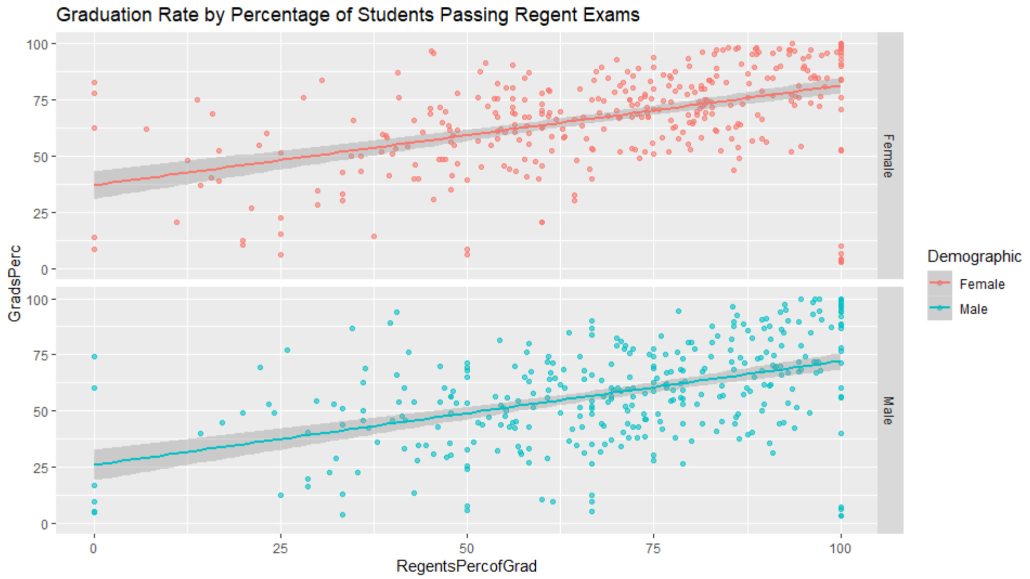
Student type was broken out by Special Education students and General education students. Special Education designation are determined by a combination of factors including test scores and professional evaluations.



Bob R. Gardner - R programming Class - 12/17/19

By Gender

Independent Variable	Dependent Variable(s)	Fit (Adjusted R^2)	Mean Square Errors
Graduation Percentage	Percentage of Cohort Passing Regents	0.2036	490.8031
	Drop Out Percentage	0.364	325.7631
	Percentage of Cohort Passing Regents & Drop Out Percentage	0.4326	284.5956
	Dummy Variable		4429.988



Gender was separated into male and female.

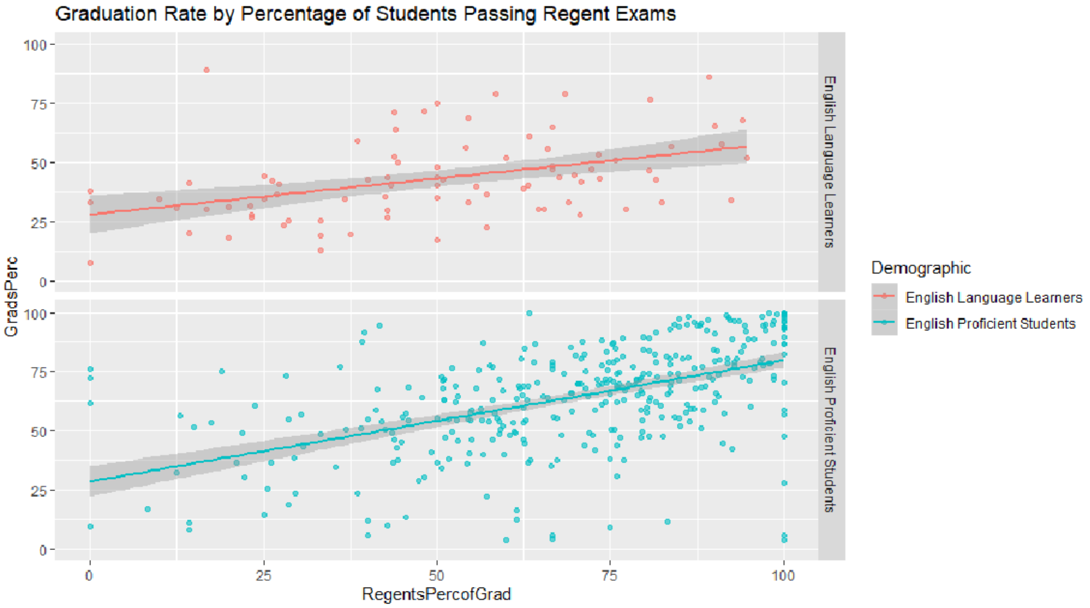
Bob R. Gardner - R programming Class - 12/17/19

By English Proficiency

Independent Variable	Dependent Variable(s)	Fit (Adjusted R^2)	Mean Square Errors
Graduation Percentage	Percentage of Cohort Passing Regents	0.3119	352.5415
	Drop Out Percentage	0.4183	302.174
	Percentage of Cohort Passing Regents & Drop Out Percentage	0.5147	248.0603
	Dummy Variable		4132.885

English Proficiency is determined by English Language Learners (ELL) and English proficient Learners.

ELL designation is determined by testing and professional evaluation.



Model Fit

Each model was had three factors judging it. The first was the Adjusted R^2 which is a fit of how well the model fits the data points and Mean Square Errors which measure how far of each “prediction” was off from actual values. The model took the existing raw data, made a “prediction” and compared that information to the actual data.

In each case with more dependent variables added, the model improved. The best R^2 was student type by approx. 0.6 and the best Mean Square Error (smallest value) was race by approx. 223.

However, the there is a third factor determining the data which is Hypothesis Testing. For each model there was a hypothesis test as to whether the “predicted” value and the actual value were statistically different from each other. The confidence interval was 95% meaning in the long run there is 5% rate of choosing the wrong hypothesis.

Summary of that data is below by p-values. Any p value greater than 0.05 is the cut off mark for determining whether the prediction and actual data were statistically different.

Model	Probability values
Race	0.8589
Student Type	0.0001
English Proficiency	0.004619
Gender	0.5004

In light of the hypothesis test, Race has the strongest probability value with p value = 0.8589 and range in difference between the actual and predicted between -2.13 and 1.77. As a result the Race model is best case in determining in predicting graduation rates by cohort.

Model Interpretation.

In terms of the regression model. The equation is

$y = 57.9 + 02778x1 -1.41912x2$

where x1 is the factor of the percentage of students passing the regents as a percentage of cohorts and

x2 is drop out factor – students dropping out of high school as percentage of cohorts.

X2 is a negative coefficient due to the negative relationship that drop out rates have with graduation rates.

In conclusion

In looking at over 2,000 records of data for graduating class between 2005 and 2010 (high school students) the race model was determined to be the best predictor of graduation rate.

Appendix: R Code

```
# Getting Data ready
raw = read.csv("cohort.csv")
library(dplyr)

rawselect = raw %>% select(Demographic, DBN, Cohort, Total.Cohort,
                           Total.Grads...n, Total.Grads.....of.cohort,
                           Total.Regents...n, Total.Regents.....of.cohort,
                           Total.Regents.....of.grads,
                           Dropped.Out.....of.cohort) %>%
                           filter(Total.Regents...n !="s" & Cohort == 2005)

colnames(rawselect) = c("Demographic", "DBN", "Year", "Cohortn", "Gradsn",
                        "GradsPerc","Regentsn","RegentsPercofCohort",
                        "RegentsPercofGrad","DropOutPerc")
main = rawselect %>% select(Demographic, DBN, Year,
                            Gradsn, GradsPerc,
                            Regentsn, RegentsPercofCohort, RegentsPercofGrad,
                            DropOutPerc) %>%
                            mutate(Borough = substr(DBN,3,3)) %>%
                            mutate(District = substr(DBN,1,2)) %>%
                            mutate(Cohortn = as.numeric(rawselect$Cohortn))
glimpse(main)

#******************************************************************

ClassType = c("General Education Students","Special Education Students")
Race = c("Asian","Black","White","Hispanic")
Gender = c("Female","Male")
Language = c("English Language Learners","English Proficient Students")

mainClassType = main %>% filter(Demographic %in% ClassType)
mainRace = main %>% filter(Demographic %in% Race)
mainGender = main %>% filter(Demographic %in% Gender)
mainLanguage = main %>% filter(Demographic %in% Language)

##new models
#ModelRace1
modelRace1 = lm(GradsPerc~RegentsPercofGrad, data = mainRace)
modelRace1
summary(modelRace1)
GPP = predict(modelRace, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelRace2
modelRace2 = lm(GradsPerc~DropOutPerc, data = mainRace)
modelRace2
summary(modelRace2)
GPP = predict(modelRace2, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelRace3
modelRace3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data = mainRace)
modelRace3
summary(modelRace3)
GPP = predict(modelRace3, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType1
modelClassType1= lm(GradsPerc~RegentsPercofGrad, data = mainClassType)
modelClassType1
summary(modelClassType1)
GPP = predict(modelClassType, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType2
modelClassType2 = lm(GradsPerc~DropOutPerc, data = mainClassType)
modelClassType2
summary(modelClassType2)
GPP = predict(modelClassType2, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType3
modelClassType3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainClassType)
modelClassType3
summary(modelClassType3)
GPP = predict(modelClassType3, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender1
modelGender1 = lm(GradsPerc~RegentsPercofGrad, data = mainGender)
modelGender1
summary(modelGender1)
GPP = predict(modelGender, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender2
modelGender2 = lm(GradsPerc~DropOutPerc, data = mainGender)
modelGender2
summary(modelGender2)
GPP = predict(modelGender2, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender3
modelGender3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data = mainGender)
modelGender3
summary(modelGender3)
GPP = predict(modelGender3, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage1
modelLanguage1= lm(GradsPerc~RegentsPercofGrad, data = mainLanguage)
modelLanguage1
summary(modelLanguage1)
GPP = predict(modelLanguage1, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage2
modelLanguage2 = lm(GradsPerc~DropOutPerc, data = mainLanguage)
modelLanguage2
summary(modelLanguage2)
GPP = predict(modelLanguage2, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage3
modelLanguage3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainLanguage)
modelLanguage3
summary(modelLanguage3)
GPP = predict(modelLanguage3, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage.Dummy
modelLanguage.Dummy = lm(GradsPerc~-1,data = mainLanguage)
modelLanguage.Dummy
summary(modelLanguage.Dummy)
GPP = predict(modelLanguage.Dummy, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelRace.Dummy
modelRace.Dummy = lm(GradsPerc~-1,data = mainRace)
modelRace.Dummy
summary(modelRace.Dummy)
GPP = predict(modelRace.Dummy, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType.Dummy
modelClassType.Dummy = lm(GradsPerc~-1,data = mainClassType)
modelClassType.Dummy
summary(modelClassType.Dummy)
GPP = predict(modelClassType.Dummy, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender.Dummy
modelGender.Dummy = lm(GradsPerc~-1,data = mainGender)
modelGender.Dummy
summary(modelGender.Dummy)
GPP = predict(modelGender.Dummy, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

##new models
#ModelRace.T1
modelRace.T1 = rpart(GradsPerc~RegentsPercofGrad, data = mainRace)
GPP = predict(modelRace.T1, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelRace.T2
modelRace.T2 = rpart(GradsPerc~DropOutPerc, data = mainRace)
GPP = predict(modelRace.T2, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelRace.T3
modelRace.T3 = rpart(GradsPerc~RegentsPercofGrad+DropOutPerc, data = mainRace)
modelRace.T3
summary(modelRace.T3)
GPP = predict(modelRace.T3, newdata = mainRace)
mean((mainRace$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType.T1
modelClassType.T1= rpart(GradsPerc~RegentsPercofGrad, data = mainClassType)
GPP = predict(modelClassType.T1, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType.T2
modelClassType.T2 = rpart(GradsPerc~DropOutPerc, data = mainClassType)
GPP = predict(modelClassType.T2, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelClassType.T3
modelClassType.T3 = rpart(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainClassType)
GPP = predict(modelClassType.T3, newdata = mainClassType)
mean((mainClassType$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender.T1
modelGender.T1 = rpart(GradsPerc~RegentsPercofGrad, data = mainGender)
GPP = predict(modelGender.T1, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender.T2
modelGender.T2 = rpart(GradsPerc~DropOutPerc, data = mainGender)
GPP = predict(modelGender.T2, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelGender.T3
modelGender.T3 = rpart(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainGender)
GPP = predict(modelGender.T3, newdata = mainGender)
mean((mainGender$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage.T1
modelLanguage.T1= rpart(GradsPerc~RegentsPercofGrad, data = mainLanguage)
modelLanguage.T1
summary(modelLanguage.T1)
GPP = predict(modelLanguage.T1, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage.T2
modelLanguage.T2 = rpart(GradsPerc~DropOutPerc, data = mainLanguage)
GPP = predict(modelLanguage.T2, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)

#ModelLanguage.T3
modelLanguage.T3 = rpart(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainLanguage)
GPP = predict(modelLanguage.T3, newdata = mainLanguage)
mean((mainLanguage$GradsPerc-GPP)^2, na.rm = TRUE)


#**************
ggplot(mainRace) +
  aes(x = RegentsPercofGrad, y = GradsPerc, color = Demographic) +
  geom_point(alpha = 0.6)+geom_smooth(method = "lm")+
  facet_grid(Demographic ~ .) +
  ggtitle("Graduation Rate by Percentage of Students Passing Regent Exams")

ggplot(mainClassType) +
  aes(x = RegentsPercofGrad, y = GradsPerc, color = Demographic) +
  geom_point(alpha = 0.6)+geom_smooth(method = "lm")+
  facet_grid(Demographic ~ .) +
  ggtitle("Graduation Rate by Percentage of Students Passing Regent Exams")

ggplot(mainLanguage) +
  aes(x = RegentsPercofGrad, y = GradsPerc, color = Demographic) +
  geom_point(alpha = 0.6)+geom_smooth(method = "lm")+
  facet_grid(Demographic ~ .) +
  ggtitle("Graduation Rate by Percentage of Students Passing Regent Exams")

ggplot(mainGender) +
  aes(x = RegentsPercofGrad, y = GradsPerc, color = Demographic) +
  geom_point(alpha = 0.6)+geom_smooth(method = "lm")+
  facet_grid(Demographic ~ .) +
  ggtitle("Graduation Rate by Percentage of Students Passing Regent Exams")
```

```
##t-test

#ModelRace3
modelRace3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data = mainRace)
GPP = predict(modelRace3, newdata = mainRace)
t.test(mainRace$GradsPerc, GPP, data = mainRace)


#ModelClassType3
modelClassType3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainClassType)
GPP = predict(modelClassType3, newdata = mainClassType)
t.test(mainRace$GradsPerc, GPP, data = mainClassType)

#ModelLanguage3
modelLanguage3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data =
mainLanguage)
GPP = predict(modelLanguage3, newdata = mainLanguage)
t.test(mainRace$GradsPerc, GPP, data = mainLanguage)

#ModelGender3
modelGender3 = lm(GradsPerc~RegentsPercofGrad+DropOutPerc, data = mainGender)
GPP = predict(modelGender3, newdata = mainGender)
t.test(mainRace$GradsPerc, GPP, data = mainGender)
```