Problem Set 8: Prediction (Answer Key)

Joe Ornstein

Due November 18, 2020

First, load the labeled training data.

```
data <- read_csv('data/CCES-Train-POLS-7012.csv') %>%
  mutate(age = 2018 - birthyr,
         id = 1:n(),
         immstat = factor(immstat),
         faminc_new = factor(faminc_new))
```

Split into a training set and a test set.

```
train <- data %>%
  sample_frac(0.7)

test <- data %>%
  anti_join(train, by = 'id')
```

Then fit some models on the `train` set:

```
library(kknn)

model1 <- lm(democratic2016 ~ region + gender + educ + race +
               pew_religimp + religpew + urbancity + age,
             data = train)

model2 <- kknn(democratic2016 ~ region + gender + educ + race +
               pew_religimp + religpew + urbancity + age,
             train = train,
             test = test)

# kitchen sink lm
model3 <- lm(democratic2016 ~ .,
             data = train)

# lm keeping only the strongest predictor variables
model4 <- lm(democratic2016 ~ region + gender + sexuality +
               educ + race + employ + pew_religimp + religpew +
               urbancity + milstat_5,
             data = train)

# logistic
model5 <- glm(democratic2016 ~ region + gender + sexuality +
               educ + race + employ + pew_religimp + religpew +
               urbancity + milstat_5,
             data = train,
```

```
              family = 'binomial')
```

Which does best predicting the training set?

```
# function to compute classification accuracy
classification_accuracy <- function(truth, predicted){
  predicted <- ifelse(predicted > 0.5, 1, 0)
  sum(truth == predicted) / length(truth) * 100
}

classification_accuracy(truth = test$democratic2016,
                        predicted = predict(model1, test))
```

```
## [1] 75.43333
```

```
classification_accuracy(truth = test$democratic2016,
                        predicted = model2$fitted.values)
```

```
## [1] 70.26667
```

```
classification_accuracy(truth = test$democratic2016,
                        predicted = predict(model3, test))
```

```
## [1] 75.9
```

```
classification_accuracy(truth = test$democratic2016,
                        predicted = predict(model4, test))
```

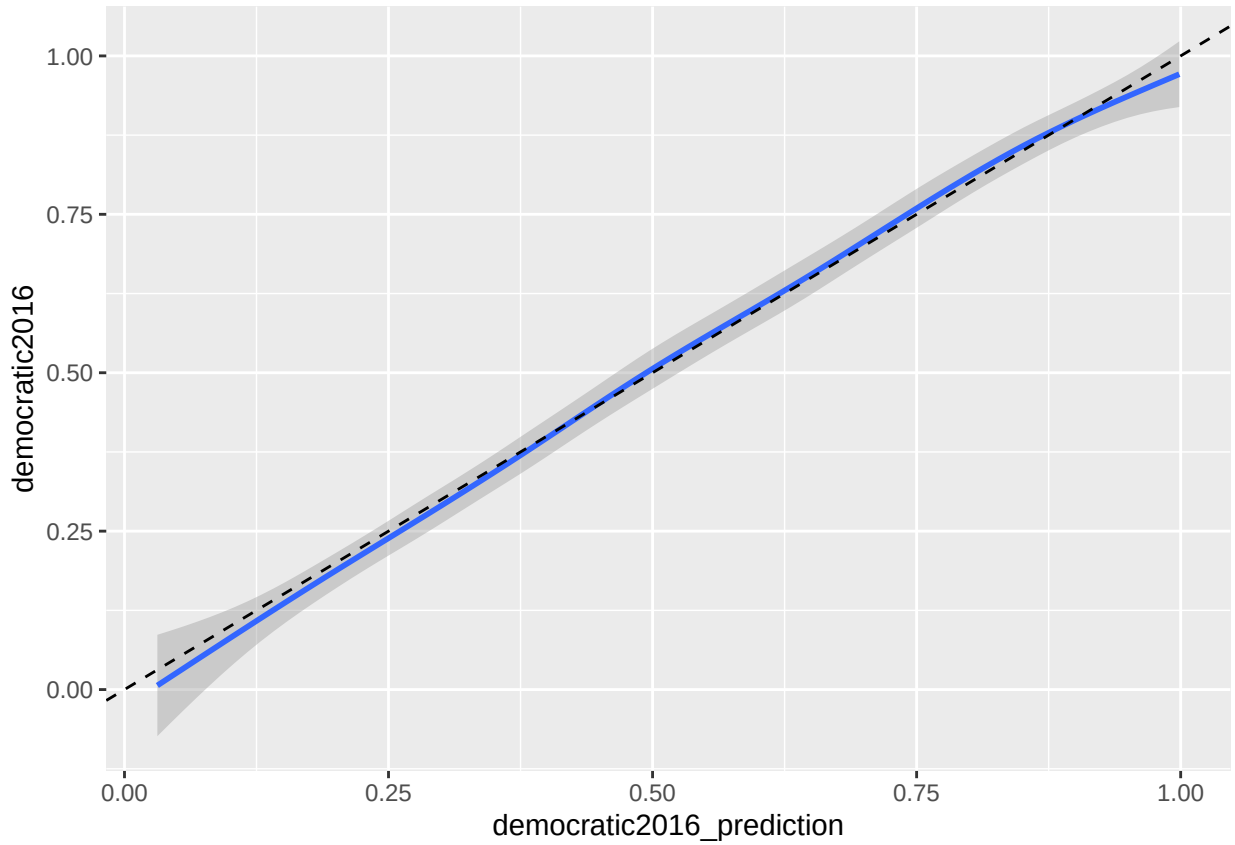
```
## [1] 76.13333
```

```
classification_accuracy(truth = test$democratic2016,
                        predicted = predict(model5, test, type = 'response'))
```

```
## [1] 76.16667
```

The kitchen sink is pretty good, but removing some of the least significant variables does better. A logistic model (see the slides; it forces the prediction between zero and one) does even better. Plot the fit:

```
test %>%
  mutate(democratic2016_prediction = predict(model5,
                                             test, type = 'response')) %>%
  ggplot() +
  geom_smooth(aes(x=democratic2016_prediction, y=democratic2016)) +
  geom_abline(intecept = 0, slope = 1, linetype = 'dashed')
```



Now make predictions on the test set:

```
data <- read_csv('data/CCES-Test-POLS-7012.csv') %>%
  mutate(age = 2018 - birthyr,
         id = 1:n())

data <- data %>%
  mutate(p_democrat = predict(model5, data, type = 'response'))

write_csv(data, 'submissions/ornstein.csv')
```