Harvard PH125.9x Capstone Project (Choose Your Own)

Michael Graber

11/16/2020

Introduction

As part of the Data Science: Capstone course in the Data Science Professional Certificate EdX program, students are encouraged to find a data set of their own or from an online source to build a machine learning model. The intent of the project is to demonstrate the ability to analyze a data set as well as clearly communicate the process and insights gained from the analysis.

This project intends to develop a model to predict the compressive strength of a given concrete mixture after a certain number of days. At first, I considered this dataset mundane and flat. Upon reflection, given how useful concrete is in both developed and developing nations, the ability to use local materials and know that the concrete will be strong seems to add value to the data and the utility of a potential model.

Objectives & Approach

The objective of this analysis is to define a reusable model to predict the compressive strength of a concrete mixture comprised of up to 7 ingredients and allowed to cure for a certain number of days before strength measurement. I will use several approaches to identify models and determine which model has the highest accuracy for predicting results.

The data set has 9 variables, 8 dependent and one independent. All variables are numeric. For purposes of brevity, I will establish shortened attribute names for the original variable names given by the dataset:

Table 1: Original and Shortened Attribute Names

Original Attribute Name	Shortened Name
Cement (component 1)(kg in a m^3 mixture)	Cement
Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Slag
Fly Ash (component 3)(kg in a m^3 mixture)	Ash
Water (component 4)(kg in a m^3 mixture)	Water
Superplasticizer (component 5)(kg in a m^3 mixture)	Plasticizer
Coarse Aggregate (component 6)(kg in a m^3 mixture)	Coarse_Agg
Fine Aggregate (component 7)(kg in a m^3 mixture)	Fine_Agg
Age (day)	Days
Concrete compressive strength(MPa, megapascals)	Strength

Not every blend utilizes all seven component ingredients. There are some which use no quantity of Slag, Ash, or Plasticizer. All blends use some amount of cement, and all blends use water (which is needed to trigger the chemical reaction to re-form the ingredients into concrete.)

The range of curing days includes a minimum of 1 day and a maximum of 365 days. I also note that the median days of curing is 28 days, which is acknowledged as an industry standard for measuring compressive strength in civil engineering. While 28 days is not a magic number - as mentioned above, application requirements and strength thresholds vary - it is used largely to help keep construction projects on schedule.

Within the dataset, there is a large variety of concrete “blends” as well as several records for the same blend

Table 2: Summary of Component Attributes

	Cement	Slag	Ash	Water	Plasticizer	Coarse_Agg	Fine_Agg
	Min. :102.0	Min. : 0.0	Min. : 0.00	Min. :121.8	Min. : 0.000	Min. : 801.0	Min. :594.0
	1st Qu.:192.4	1st Qu.: 0.0	1st Qu.: 0.00	1st Qu.:164.9	1st Qu.: 0.000	1st Qu.: 932.0	1st Qu.:731.0
	Median :272.9	Median : 22.0	Median : 0.00	Median :185.0	Median : 6.350	Median : 968.0	Median :779.5
	Mean :281.2	Mean : 73.9	Mean : 54.19	Mean :181.6	Mean : 6.203	Mean : 972.9	Mean :773.6
	3rd Qu.:350.0	3rd Qu.:142.9	3rd Qu.:118.27	3rd Qu.:192.0	3rd Qu.:10.160	3rd Qu.:1029.4	3rd Qu.:824.0
	Max. :540.0	Max. :359.4	Max. :200.10	Max. :247.0	Max. :32.200	Max. :1145.0	Max. :992.6

Table 3: Summary of Curing Durations (Days)

	Days
	Min. : 1.00
	1st Qu.: 7.00
	Median : 28.00
	Mean : 45.66
	3rd Qu.: 56.00
	Max. :365.00

measured after different curing durations. Taking a brief sample, we can see 10 records with 500kg/m3 of Cement with varying amounts if Coarse Aggregate, and 8 records with the exact same combination of ingredients yet aged for an increasing number of days, resulting in increasing compressive strength:

Table 4: Sample of Data (with Cement = 500)

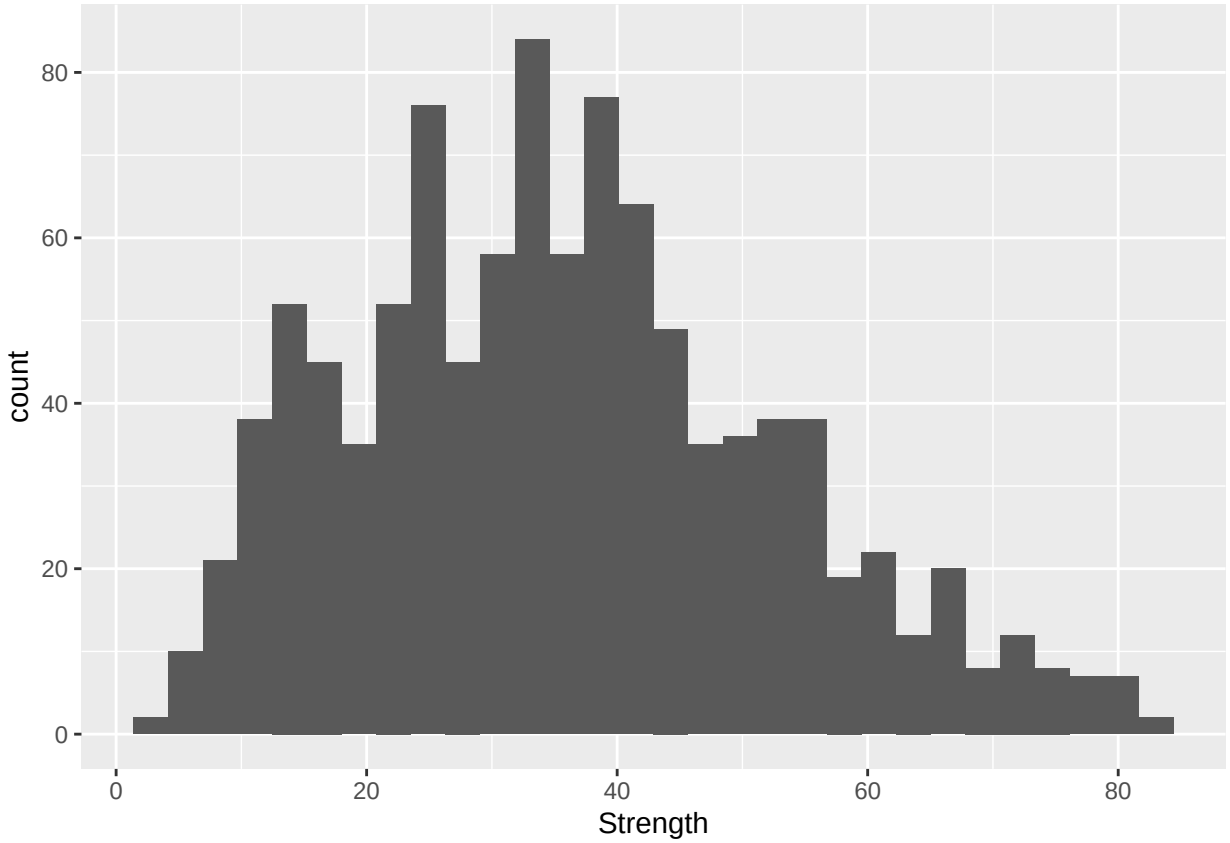
Cement	Slag	Ash	Water	Plasticizer	Coarse_Agg	Fine_Agg	Days	Strength
500	0	0	140	4	966	853	28	67.56865
500	0	0	151	9	1033	655	28	69.83702
500	0	0	200	0	1125	613	1	12.63810
500	0	0	200	0	1125	613	3	26.06219
500	0	0	200	0	1125	613	7	33.21206
500	0	0	200	0	1125	613	14	36.93523
500	0	0	200	0	1125	613	28	44.09199
500	0	0	200	0	1125	613	90	47.22221
500	0	0	200	0	1125	613	180	51.04191
500	0	0	200	0	1125	613	270	55.15808

Methods

Multiple Linear Regression Model

To see if a linear regression model is practicable, we need to verify that the dependent variable approximates a normal distribution. We can do that with a quick plot:

```
ggplot(data = Concrete_Data, aes(x=Strength)) + geom_histogram(bins = 30)
```



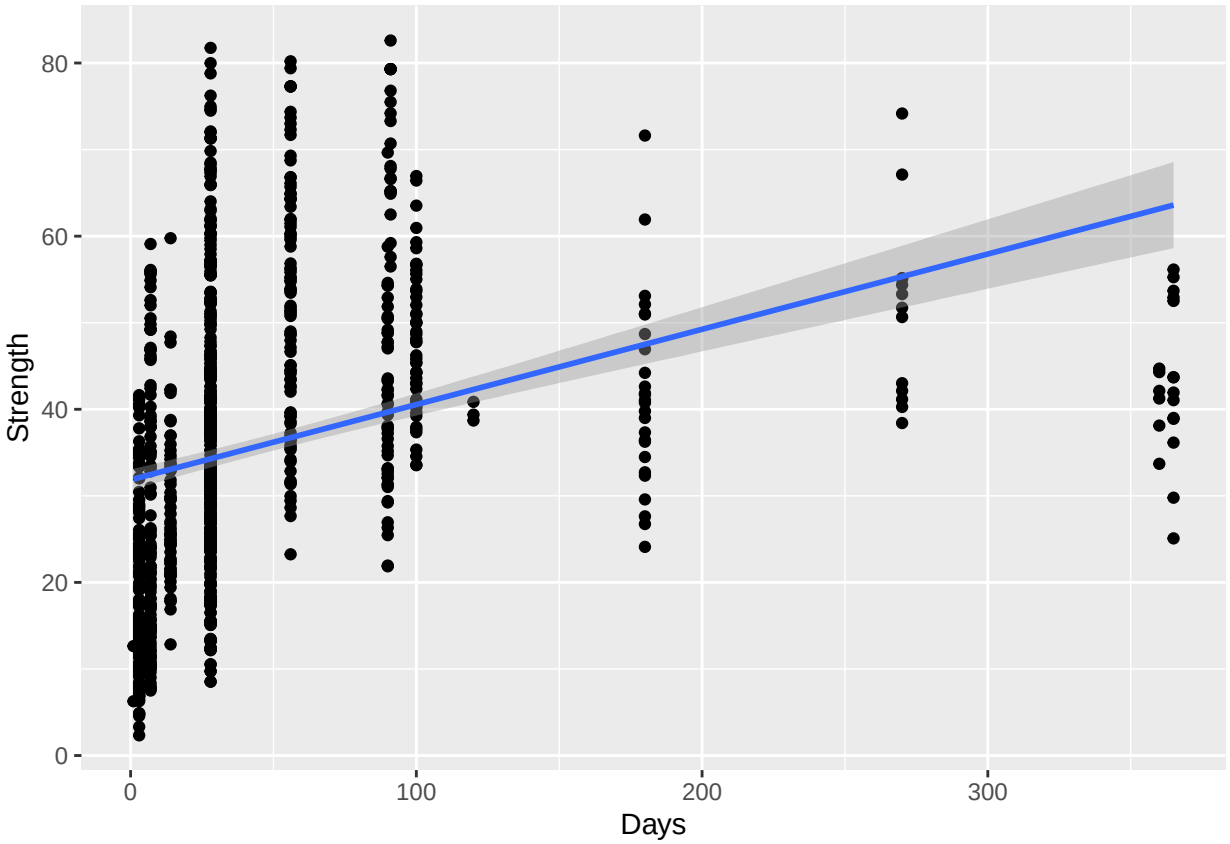
Let's compare the relationship of age (Days) and Strength. The data seems pre-stratified by virtue of the ages at which the strength is measured:

```
Concrete_Data %>% summarize(r = cor(Days, Strength)) %>% pull(r)
```

```
## [1] 0.328877
```

```
ggplot(data = Concrete_Data, aes(Days, Strength)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Based on the graph above, we can see in general that the longer a given concrete mixture is permitted to age, the stronger it gets.

Beyond curing time, other aspects of the concrete mixture affect the compressive strength. A multiple linear regression model might provide a good place to start to build a predictive solution:

```
linear_model <- train(Strength ~ ., data = Concrete_Data, method = "lm")
summary(linear_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.653  -6.303   0.704   6.562  34.446
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -23.163756  26.588421  -0.871 0.383851
## Cement        0.119785   0.008489  14.110  < 2e-16 ***
## Slag          0.103847   0.010136  10.245  < 2e-16 ***
## Ash           0.087943   0.012585   6.988 5.03e-12 ***
## Water        -0.150298   0.040179  -3.741 0.000194 ***
## Plasticizer   0.290687   0.093460   3.110 0.001921 **
## Coarse_Agg    0.018030   0.009394   1.919 0.055227 .
## Fine_Agg      0.020154   0.010703   1.883 0.059968 .
## Days          0.114226   0.005427  21.046  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 10.4 on 1021 degrees of freedom
## Multiple R-squared:  0.6155, Adjusted R-squared:  0.6125
## F-statistic: 204.3 on 8 and 1021 DF,  p-value: < 2.2e-16
```

The results of this first linear model show most coefficients have a low P-value, with two attributes on the threshold (Coarse_Agg and Fine_Agg are both approximately 0.06), and a relatively high P-value for the Y-intercept (0.38). Overall, this model only explains about 61% of the variability of the dependent value, compressive Strength. This linear regression model results in a predictive formula of:

Strength = 0.12 * Cement + 0.104 * Slag + 0.088 * Ash - 0.15 * Water + 0.29 * Plasticizer + 0.018 * Coarse_Agg + 0.02 * Fine_Agg + 0.114 * Days - 23.163

I think we can do better by looking further at the attributes, their correlation to the strength of the concrete, and their interaction related to the actual chemical reaction that occurs during the curing phase.

Correlation Coefficients

Interaction considerations

Since some of the mixtures include certain components and not others, we have to consider whether or not the components are part of the chemical reaction (hydration), part of the final structural (compressive) strength, or neither. According to industry publications, cement, ash, and slag are considered "cementitious" components and contribute to the hydration reaction of the curing concrete. Plasticizer, likewise, allows the production of concrete with less water, and slow the curing of concrete (as would adding more water and prolonging the drying/curing process.) Other ingredients, such as coarse and fine aggregate, are necessary for the ultimate strength of the cured mixture, but do not play a part in the chemistry.

As part of this data set review and model, we have to consider whether to include in the correlation analysis the zero values for cementitious components: If they are not present, they cannot influence the chemical reaction. In one sense, we are treating these attributes as both logistic and continuous. For the high level perspective of determine a correlation coeffecient between these attributes and the Strength attribute, I have filtered out the zero values.

As an example, we can see that with or without the Ash=0 values, the correlation of Ash content to Strength is negative, but more pronounced if the zero values are excluded. This indicates that stronger concrete uses less Ash, and the strongest (regardless of curing time) uses none.

Correlation Coefficient of Ash:Strength with all values

```
Concrete_Data %>%
  summarize(r = cor(Ash, Strength)) %>%
  pull(r)
```

```
## [1] -0.1057533
```

```
# Correlation Coefficient of Ash:Strength, ignoring mixtures with zero Ash
Concrete_Data %>%
  filter(Ash > 0) %>%
  summarize(r = cor(Ash, Strength)) %>%
  pull(r)
```

```
## [1] -0.2315816
```

For visual review and for the purposes of considering variable interactions, I will exclude the zero values.

```
colorGray = "#666666"

Concrete_Data %>%
  summarize(r = cor(Cement, Strength)) %>%
  pull(r)
```

```
## [1] 0.4978327
```

```
Concrete_Data %>%
  filter(Slag > 0) %>%
  summarize(r = cor(Slag, Strength)) %>%
  pull(r)
```

```
## [1] -0.08614059
```

```
Concrete_Data %>%
  filter(Plasticizer > 0) %>%
  summarize(r = cor(Plasticizer, Strength)) %>%
  pull(r)
```

```
## [1] 0.2845361
```

```
Concrete_Data %>%
  summarize(r = cor(Water, Strength)) %>%
  pull(r)
```

```
## [1] -0.2896135
```

```
Concrete_Data %>%
  summarize(r = cor(Days, Strength)) %>%
  pull(r)
```

```
## [1] 0.328877
```

```
Concrete_Data %>%
  summarize(r = cor(Coarse_Agg, Strength)) %>%
  pull(r)
```

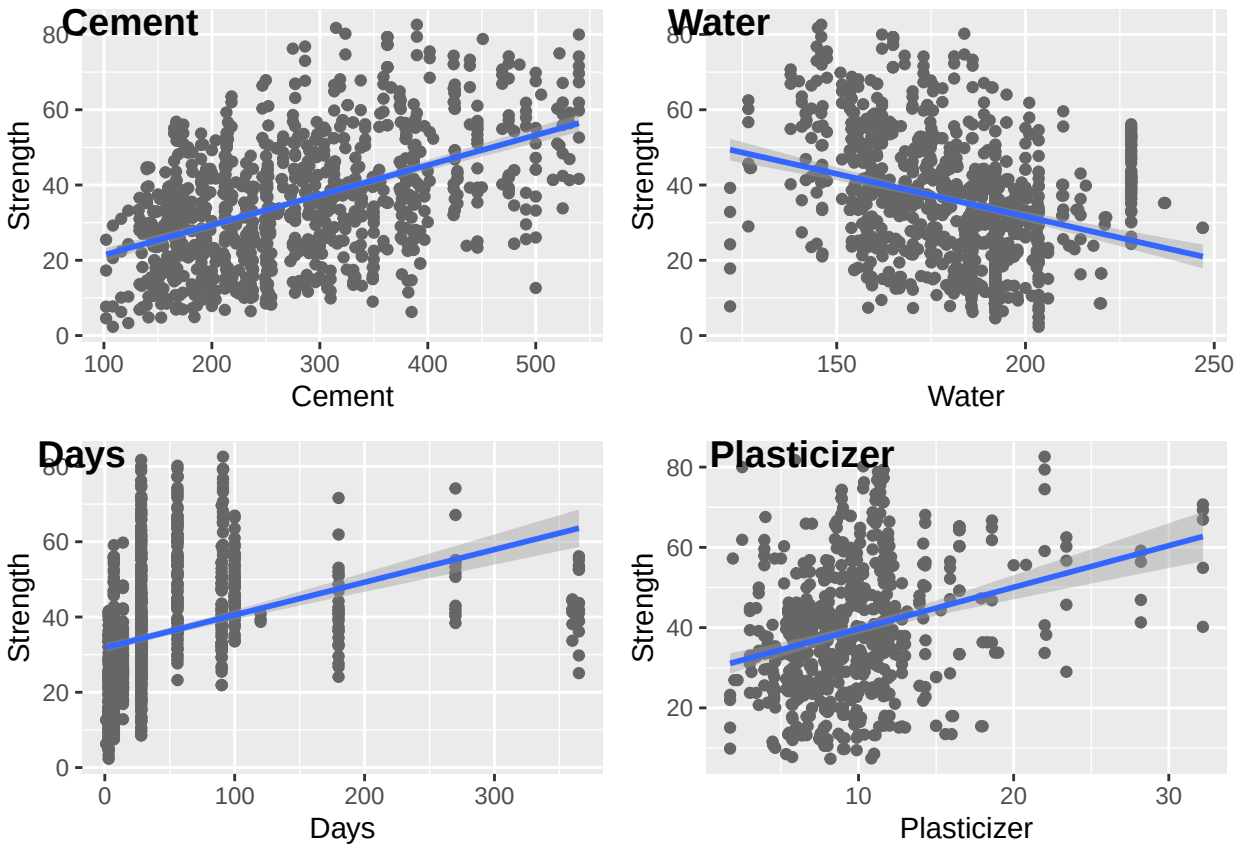
```
## [1] -0.1649278
```

```
Concrete_Data %>%
  summarize(r = cor(Fine_Agg, Strength)) %>%
  pull(r)
```

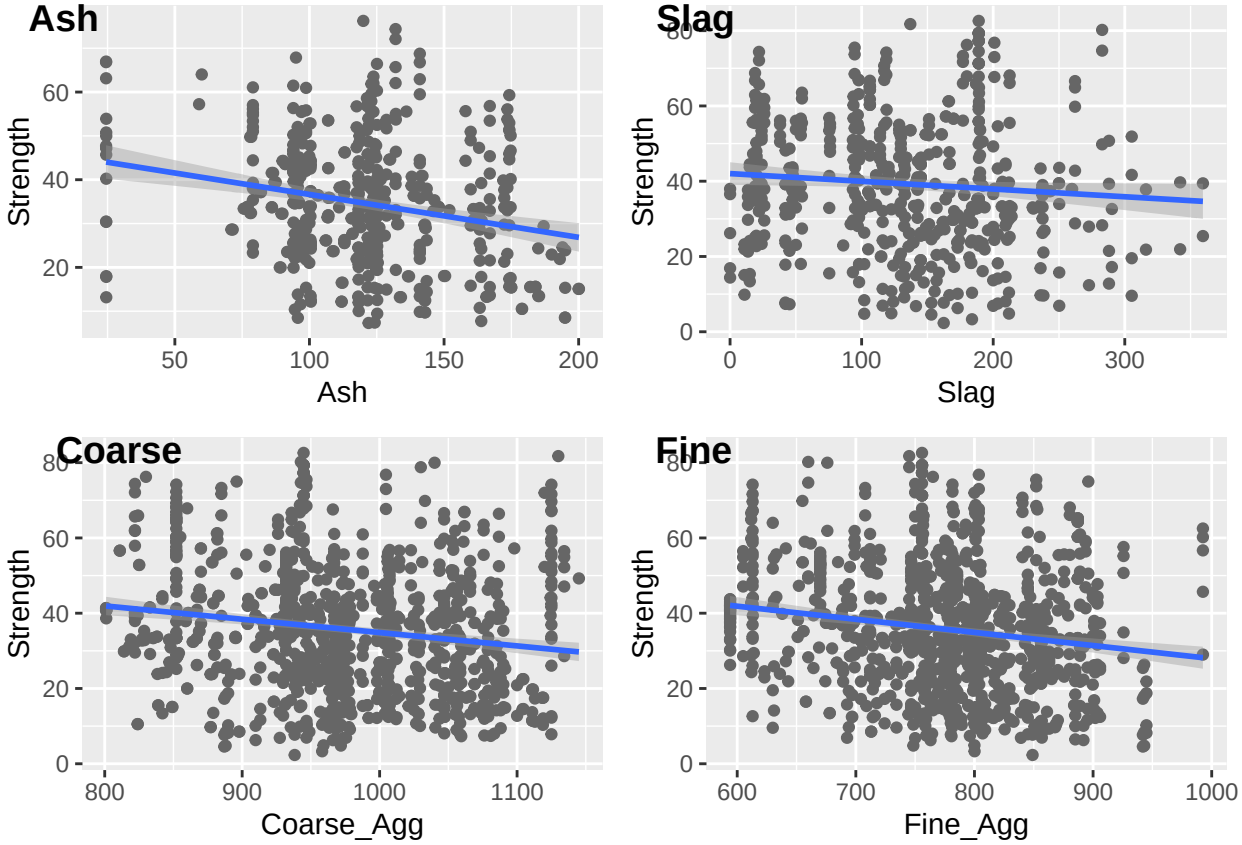
```
## [1] -0.167249
```

```
gpCorCement <- ggplot(data = Concrete_Data,
                      aes(Cement, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorAsh <- ggplot(data = (Concrete_Data %>% filter(Ash > 0)),
                   aes(Ash, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorSlag <- ggplot(data = (Concrete_Data %>% filter(Slag > 0)),
                    aes(Slag, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorPlasticizer <- ggplot(data = (Concrete_Data %>% filter(Plasticizer > 0)),
                           aes(Plasticizer, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorWater <- ggplot(data = Concrete_Data,
                     aes(Water, Strength)) +
```

```
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorDays <- ggplot(data = Concrete_Data,
                    aes(Days, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorCoarse <- ggplot(data = Concrete_Data,
                      aes(Coarse_Agg, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

gpCorFine <- ggplot(data = Concrete_Data,
                    aes(Fine_Agg, Strength)) +
  geom_point(colour = colorGray) +
  geom_smooth(method = 'lm')

# Draw some plots for discussion
ggarrange(gpCorCement, gpCorWater, gpCorDays, gpCorPlasticizer,
          labels = c("Cement", "Water", "Days", "Plasticizer"),
          ncol = 2, nrow = 2)
```




```
ggarrange(gpCorAsh, gpCorSlag, gpCorCoarse, gpCorFine,
          labels = c("Ash", "Slag", "Coarse", "Fine"),
          ncol = 2, nrow = 2)
```



As the plots above show, the solids that eventually comprise the concrete after curing all have similar slopes and likely have no interaction among them. However, there are other interactions to consider: The correlation coefficient for water is negative, while those for days of curing and plasticizer are positive. There may be interactions worth evaluating. Likewise, the coefficient for cement is positive, while those for all other solids are negative. We will explore these interactions as ratios in further regression models.

Water / Days

One interaction for consideration - based on both intuition and on the respective coefficients of correlation is that of Water and Days. That is, does the interaction of these two attributes taken together improve the model as compared to the two attributes considered separately?

```
Conc_Interact2 <- Concrete_Data %>%
  mutate(WaterDays = Water/Days)
linear_model2 <- train(Strength ~ Cement + Slag + Ash + Plasticizer + Coarse_Agg + Fine_Agg +
                         WaterDays, data = Conc_Interact2, method = "lm")
summary(linear_model2)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.428  -5.987  -1.026   5.386  48.059
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -93.435664   8.669930 -10.777  < 2e-16 ***
## Cement        0.146328   0.005037  29.049  < 2e-16 ***
```

```
## Slag          0.119954   0.006144  19.523  < 2e-16 ***
## Ash           0.098591   0.008147  12.101  < 2e-16 ***
## Plasticizer   0.176985   0.067006   2.641  0.00838 **
## Coarse_Agg    0.043502   0.004447   9.783  < 2e-16 ***
## Fine_Agg      0.048240   0.005224   9.235  < 2e-16 ***
## WaterDays    -0.449047   0.013662 -32.868  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.697 on 1022 degrees of freedom
## Multiple R-squared:  0.7308, Adjusted R-squared:  0.729
## F-statistic: 396.3 on 7 and 1022 DF,  p-value: < 2.2e-16
```

In this updated model, we can see that the Standard Error for the Intercept and the overall Residual Standard Error for the model have both improved. Also, the multiple R-squared value for the model as a whole now demonstrates that this model explains about 73% of the variation in the dependent variable - an improvement over the original.

Cement + Ash

Another consideration might be the combination of Portland Cement and Fly Ash. Both contribute to the chemical hardening of concrete during the curing process. We can adjust the model to combine them into one attribute; we can add an interaction to the model:

```
Conc_Interact3 <- Conc_Interact2 %>% mutate(Silicates = Cement+Ash)
linear_model3 <- train(Strength ~ Silicates + Slag + Plasticizer + Coarse_Agg +
                         Fine_Agg + WaterDays,
                       data = Conc_Interact3,
                       method = "lm")
summary(linear_model3)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.022  -5.692  -0.618   5.040  50.154
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.172e+02  8.614e+00 -13.606   <2e-16 ***
## Silicates    1.591e-01  5.039e-03  31.576   <2e-16 ***
## Slag         1.449e-01  5.744e-03  25.222   <2e-16 ***
## Plasticizer -1.215e-01  6.108e-02  -1.989   0.0469 *
## Coarse_Agg   5.011e-02  4.566e-03  10.975   <2e-16 ***
## Fine_Agg     6.177e-02  5.217e-03  11.842   <2e-16 ***
## WaterDays   -4.488e-01  1.421e-02 -31.576   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.049 on 1023 degrees of freedom
## Multiple R-squared:  0.7083, Adjusted R-squared:  0.7066
## F-statistic:   414 on 6 and 1023 DF,  p-value: < 2.2e-16
```

The results show that this interaction consideration does not improve the model, so we will not include this interaction moving forward.

Total Aggregate

```
Conc_Interact4 <- Conc_Interact2 %>% mutate(Aggregate = Coarse_Agg + Fine_Agg)
linear_model4 <- train(Strength ~ Cement + Slag + Ash + Plasticizer +
                         Aggregate + WaterDays,
                       data = Conc_Interact4,
                       method = "lm")
summary(linear_model4)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.425  -5.966  -1.085   5.388  47.988
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -92.202974   8.573383 -10.755  < 2e-16 ***
## Cement        0.144929   0.004820  30.066  < 2e-16 ***
## Slag          0.118505   0.005955  19.901  < 2e-16 ***
## Ash           0.096443   0.007832  12.315  < 2e-16 ***
## Plasticizer   0.204044   0.060747   3.359 0.000811 ***
## Aggregate     0.045145   0.004102  11.006  < 2e-16 ***
## WaterDays    -0.448230   0.013635 -32.873  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.697 on 1023 degrees of freedom
## Multiple R-squared:  0.7306, Adjusted R-squared:  0.729
## F-statistic: 462.3 on 6 and 1023 DF,  p-value: < 2.2e-16
```

This did not improve the model beyond the results above when we included the interaction of water and curing days, but it does produce almost identical summary statistics, so it does make the model simpler. This matches intuition, because while the aggregate component of the concrete does eventually contribute to the strength of the mixture, it plays no part in the chemical process, purely adding the structural stability of stone to the cohesive components produced.

One more complex interaction worth considering is the interaction of water and plasticizer with the days of curing. In concrete, plasticizer is used to reduce the amount of water used in the mixture while increasing strength and maintaining workability during the pouring (placement) of the mix.

```
Conc_Interact5 <- Conc_Interact4 %>%
  mutate(WaterPlasticDays = (Water + Plasticizer)/Days)
linear_model5 <- train(Strength ~ Cement + Slag + Ash +  Aggregate +
                         WaterPlasticDays,
                       data = Conc_Interact5,
                       method = "lm")
summary(linear_model5)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.851  -5.787  -0.850   5.192  46.860
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.087e+02  7.695e+00  -14.13   <2e-16 ***
## Cement            1.565e-01  3.981e-03   39.32   <2e-16 ***
## Slag              1.317e-01  5.041e-03   26.14   <2e-16 ***
## Ash               1.173e-01  5.935e-03   19.77   <2e-16 ***
## Aggregate         5.234e-02  3.750e-03   13.96   <2e-16 ***
## WaterPlasticDays -4.443e-01  1.321e-02  -33.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.705 on 1024 degrees of freedom
## Multiple R-squared:  0.7298, Adjusted R-squared:  0.7284
## F-statistic: 553.1 on 5 and 1024 DF,  p-value: < 2.2e-16
```

This combination makes trivial changes to the model results, so we can choose to keep it or drop it. Given a larger or different data set, this may change, but with the samples we have, the interaction does not seem to make a difference.

Linear Regression Equation

Taking the result of the linear model refinement and using the results of linear model 4, we end up with a formula to predict the compressive strength of a concrete mix:

Strength = 0.157 * Cement + 0.132 * Slag + 0.117 * Ash + 0.052 * (Coarse_Agg + Fine_Agg) - 0.444 * (Water + Plasticizer) / Days - .001

This results in a prediction value with a standard error of +/- 7.7 MPa

Another Potential Application

Beyond predicting what a given mixture's strength will be after a certain number of days, it may be useful to determine if a certain combination of components will achieve the construction standards expected of concrete. Since application requirements differ, we can choose one to develop the model, and adjust if and when the model is utilized.

Where I live in Massachussets, USA, the compressive strength requirement for residential concrete use is 2500PSI at 28 days for most applications. Using 1 pound per square inch = 6.895kPa, this converts to approximately 17.24 MPa after 28 days. By adding a qualifying attribute to the data, we can also build a logistic model to try to predict if a given concrete mixture will meet these requirements.

To consider the condition explored at the beginning of this analysis - that there may be multiple strength test measurements for the same mixture at different times - we will only consider those measurements taken when the curing period is 28 days. That is, we will consider:

1. days = 28 and strength >= 17.24 evaluates to meeting requirements
2. days = 28 and strength < 17.24 evaluates to failing requirements

Other considerations are inconclusive since during the time up to the 28th day, the concrete may not meet requirements, but is not yet expected to do so. Likewise, for this model usage, if a concrete mix will eventually reach the required strength but do so after the residential construction inspection deadline, it will not meet expectations.

First, we limit the dataset to just those records which are measured at 28 days and convert it to a matrix.

```
Conc_Quality <- Concrete_Data %>% filter(Days == 28) %>%
  mutate(MeetSpec = ifelse(Strength >= 17.24, 1, 0))
SVM_Data <- as.matrix(Conc_Quality)
```

Table 5: SVM C values with fit/error results

C	fit	error
1e-04	0.9341176	0.0658824
1e-03	0.9341176	0.0658824
1e-02	0.9341176	0.0658824
1e-01	0.9764706	0.0235294
1e+00	0.9835294	0.0164706
1e+01	0.9835294	0.0164706
1e+02	0.9835294	0.0164706
1e+03	0.9835294	0.0164706
1e+04	0.9835294	0.0164706
1e+05	0.9835294	0.0164706

Support Vector Machine Model

I will build a Support Vector Machine model and try out a range of C values (used to score the Constraint Violation Penalty) to find the value that results in the best prediction fit to actual values. This helps the R function build the model coefficients by softening constraints in the objective function being optimized. The default is 1, so we'll try 0.0001 to 10,000 by increasing order of magnitude. The results of the range of tests are in Table 5.

```
# build two data frames to store results
conf_matrix <- data.frame(Label=character(), Predict_0=numeric(),
                          Predict_1=numeric(), Total=numeric(),
                          stringsAsFactors = FALSE)
fit_test <- data.frame(C=numeric(), fit=numeric(), error=numeric())

lambda <- c(10^(-4:5))
for(l in lambda) {
  model <- ksvm(SVM_Data[,1:7], SVM_Data[,10], type = 'C-svc',
                kernel = 'vanilladot', C = l, scaled = TRUE)

  pred <- predict(model,SVM_Data[,1:7])
  #evaluate prediction against real values and store for review
  fit_ratio <- sum(pred == SVM_Data[,10]) / nrow(SVM_Data)
  fit_test[nrow(fit_test) + 1, ] = c(l, fit_ratio, model@error)

}
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

Next, we can review the range of results and select a value of C to move forward:

```
# review fit test results for range of C values in lambda
kable(fit_test, caption = "SVM C values with fit/error results")
```

The best value for C (Cost of Constraints Violation) is 1, so let's proceed with the model definition using that value:

Table 6: SVM Model Confusion Matrix

Label	Predict_0	Predict_1	Total
Actual_0	25	3	28
Actual_1	4	393	397
Total	29	396	425

```
model <- ksvm(SVM_Data[,1:7], SVM_Data[,10], type = 'C-svc',
              kernel = 'vanilladot', C = 1, scaled = TRUE)
```

```
##  Setting default kernel parameters
```

And we can compare the predicted values to the actual data to generate a confusion matrix (Table 6).

```
kable(conf_matrix, caption = "SVM Model Confusion Matrix")
```

This confusion matrix shows a model accuracy of (25+393)/425 = 98.35%, which is a measure of the true matches compared to the total records. It has a precision (accurate TRUEs) rate of 393/396 = 99%. Overall this model looks very useful in determining if a given concrete mix would meet residential use specifications.

Results

The multiple linear regression model went through four iterations to find a predictive equation that explains about 73% of the variability in the compressive strength of a concrete mixture after a give number of days. That formula is:

Strength = 0.157 * Cement + 0.132 * Slag + 0.117 * Ash + 0.052 * (Coarse_Agg + Fine_Agg) - 0.444 * (Water + Plasticizer) / Days - .001

This model is simplified for use as it aggregates two values (pun intended) since they have effectively combined identical effect on the dependent value (Strength), and is also further simplified by using an interaction effect between the amount of water in the mixture and the days the concrete is allowed to cure.

The support vector machine model was applied for a slightly different purpose as a tool to determine whether or not a given concrete mix would meet selected concrete performance requirements. As a sample, the residential construction specifications for Massachusetts, USA was selected. The dataset that was used for this model analysis was filtered based on the construction specification timing: 28 days.

Several values for constraint violation penalty were tested to find an optimum setting. The resulting model can then be used to predict whether a given mixture meets (1) or does not meet (0) the requirements. By using a confusion matrix, we can evaluate the accuracy (98.35%) and precision (99%) of the support vector model that was created.

Summary

I built two models based on this Concrete Data dataset, one multiple linear regression model using the entire data set, and another support vector machine model using the set of mixtures which were measured for compressive strength at 28 days.

Linear regression models have a tendency to overfit, we have to consider that when using the Strength formula that was developed. The number of attributes further increases the possibility of overfitting. The final attribut set, which reduced the number by two, may help alleviate that risk. The multiple linear regression model that we produced had an adjusted R-squared value of about 73%, which explains most of the variability of the dependent value. One additional consideration that may have reduced accuracy was the presence of multiple records with the same concrete mixture but measured for strength after different curing durations. Future considerations for analysis might be adding a time study element to those batches of records. Not many mixtures had the same number of measurements, so wrangling the data would be difficult.

The practical application of the support vector machine is also very useful since, given a variety of materials available in differing geographies, it is helpful to be able to predict if a certain mixture will meet building requirements. Our

SVM had a very high fit to actual values (98.35%) and the confusion matrix produced from the prediction/actual comparison also verified a high precision (99%).

References

Since my subject matter knowledge about concrete was limited, and I neither wanted to make assumptions nor trust incorrect intuition, I found these article useful. They may be useful to reviewers and readers.

Overview of concrete and common components

https://www.ccagc.org/resources/whats-the-difference-between-cement-and-concrete/

Use of plasticizer in concrete mixtures

https://en.wikipedia.org/wiki/Plasticizer#Concrete

Mass Residential Concrete Requirements

https://up.codes/viewer/massachusetts/irc-2015/chapter/4/foundations#R402.2

Data Set Source

https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength