Scout Oatman-Gaitan
April 22, 2020
Data Analytics

Assignment 7

General assignment: Predictive and Prescriptive data analytics. You should develop and validate predictive models (regression, classification, clustering – using one or more of the methods covered in class to date or one of your choosing) for ***two*** of the six (the Wine Quality contains red wine and white wine datasets) datasets below and apply them for decision purposes. Use the section numbering below for your written submission for this assignment. Include references – websites, papers, packages, data refs...

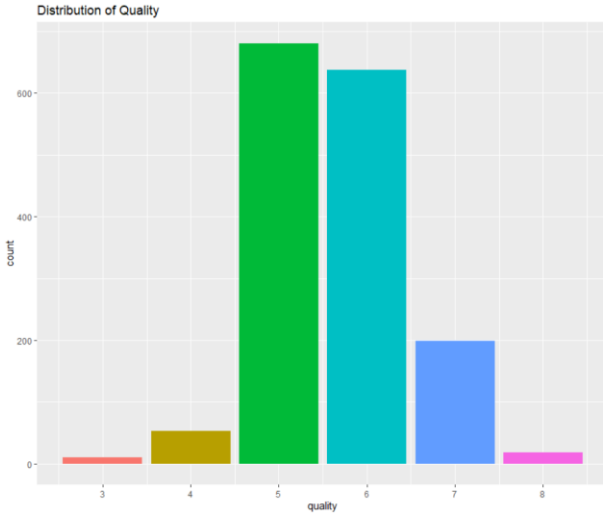
Data:

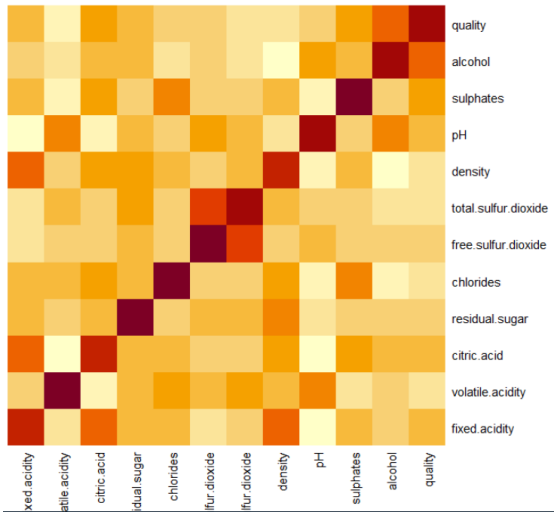
- http://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work
- http://archive.ics.uci.edu/ml/datasets/Wine+Quality (Red Wine)

1. Exploratory Data Analysis (3%) Explore the statistical aspects of both datasets. Analyze the distributions and provide summaries of the relevant statistics. Perform any cleaning, transformations, interpolations, smoothing, outlier detection/ removal, etc. required on the data. Include figures and descriptions of this exploration and a short description of what you concluded (e.g. nature of distribution, indication of suitable model approaches you would try, etc.). Min.1 page text + graphics (required).

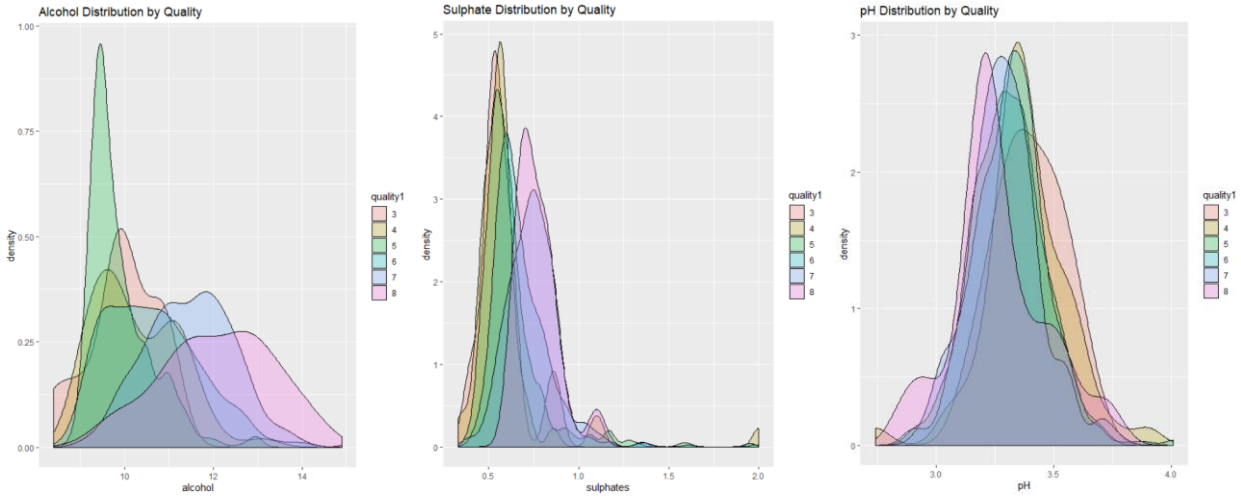
Wine

First, I did some basic analysis simply for understanding. This includes finding the dimensions, getting the summary, viewing the data, checking for null values, etc. The target variable is clearly the quality with the rest being descriptors of the wine. Here is a distribution of quality values and a simple heatmap correlation plot:





Most wines have a quality of 5 or 6. It seems that alcohol, sulphates, and pH have the most effect on quality so it makes sense to understand the distribution of quality over these three variables. Here are those plots:

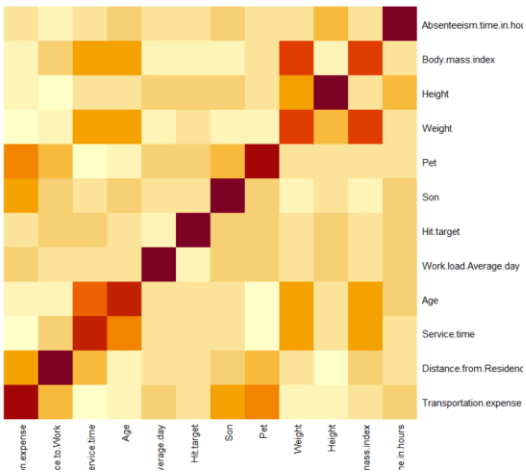


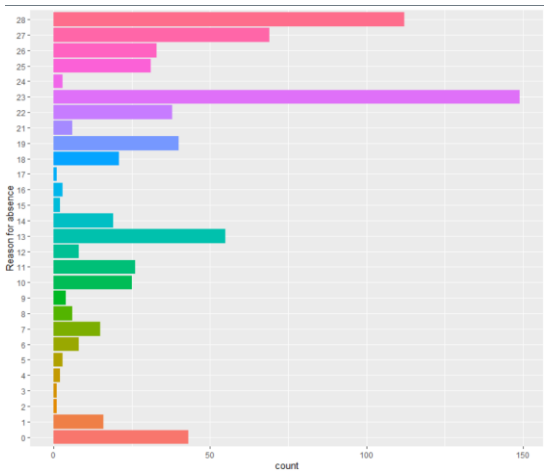
We can take away wines with higer qualitys tend tend to have more alcohol concentration. It is not true that lower quality have lower concentrations of alcohol. Higher rated wines tend to have

higher sulphate levels and it does seem that lower quality wines have lower sulphate levels. Finally it appears that highly rated wines have low pH levels and lower quality wines have higher pH levels.

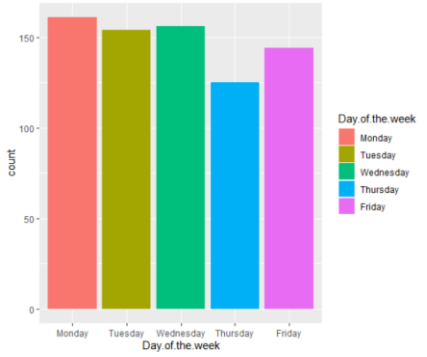
Absentee

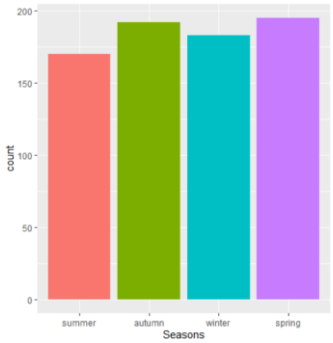
First I did some basic analysis simply for understanding. This includes finding the dimensions, getting the summary, viewing the data, checking for null values, etc. I had to assign the appropriate classes to certain columns, there are many factor types in this data. The target variable will be the number of hours absent. I assigned month names to the month columns, season names to the season, education levels to the education, and day of the week to day of the week. I made a heatmap correlation matrix below with numeric data and showed the most common reasons for missing work:

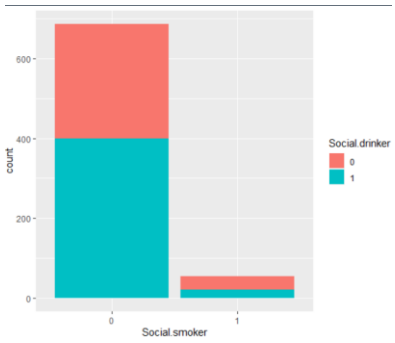




The most common reasons for missing work are 23 (medical visit) and 28 (dental visit). The most correlated to absent time is height and transportation expense. Next I looked at potential reasons people would skip – day of the week, season, social drinker/smoker, BMI, education:







These don't give us much insight as they're evenly distributed (aside from social drinker/smoker and that's because they make up the majority of employees).

2. Model Development, Validation, Optimization and Tuning (14%) Choose three (6000-level) or more different models (e.g. a model with a different set/ number of variables/ features in a regression, or classification, etc. does NOT count as a different model). Explain why you chose them. Construct the models, test/ validate them. Explain the validation approach. You can use any method(s) covered in the course. Include your code in your submission. Compare model results if applicable. Report the results of the model (fits, coefficients, graphs, trees, other measures of fit/ importance, etc.), predictors, and summary statistics. Min. 4 pages of text + graphics (required). * 4000-level will receive extra credit for 6000-level responses.

Wine

Model 1: Random Forest

Why: Random forest is a good classifier that uses a decision trees method. This problem is classifying a wine by quality given the specific attributes of the wine. Random forest will be a good model to try to predict the quality.

Construction: First, I split the data in training data and test data using 80% as the split ratio. Then I created the model using 150 trees and using all the data as attributes. Here is what the model looks like at this point:

```
> wine_rf

Call:
 randomForest(formula = quality1 ~ . - quality, data = train,
    ntree = 150)
               Type of random forest: classification
                     Number of trees: 150
No. of variables tried at each split: 3

        OOB estimate of  error rate: 30.49%
Confusion matrix:
   3 4   5   6   7 8 class.error
3 0 0   4   2   0 0   1.0000000
4 0 0  31  10   1 0   1.0000000
5 0 0 455  95   6 0   0.1816547
6 0 0 118 353  31 1   0.2982107
7 0 0   7  67  81 2   0.4840764
8 0 0   1   8   6 0   1.0000000
```

You can see that the model does well in the mid range but very poorly on the upper and lower ends, i.e. at 3, 4, and 8. The model at this point is able to understand the bulk of the data which lies in the 5-7 range but isn't good at predicting really good or really bad wine.

Validation (Approach and results): When we create a prediction using this model and compare it against the validation data we get the follow confusion matrix:

```
> confusionMatrix(rf_pred, test$quality1)
Confusion Matrix and Statistics

          Reference
Prediction   3   4   5   6   7   8
         3   0   0   0   0   0   0
         4   1   0   0   0   0   0
         5   3   6 105  39   3   0
         6   0   5  20  93  15   1
         7   0   0   0   3  24   1
         8   0   0   0   0   0   1

Overall Statistics

               Accuracy : 0.6969
                 95% CI : (0.6433, 0.7468)
    No Information Rate : 0.4219
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5121

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6
Sensitivity            0.0000 0.000000   0.8400   0.6889
Specificity            1.0000 0.996764   0.7385   0.7784
Pos Pred Value            NaN 0.000000   0.6731   0.6940
Neg Pred Value         0.9875 0.965517   0.8780   0.7742
Prevalence             0.0125 0.034375   0.3906   0.4219
Detection Rate         0.0000 0.000000   0.3281   0.2906
Detection Prevalence   0.0000 0.003125   0.4875   0.4188
Balanced Accuracy      0.5000 0.498382   0.7892   0.7336
                     Class: 7 Class: 8
Sensitivity            0.5714 0.333333
Specificity            0.9856 1.000000
Pos Pred Value         0.8571 1.000000
Neg Pred Value         0.9384 0.993730
Prevalence             0.1313 0.009375
Detection Rate         0.0750 0.003125
Detection Prevalence   0.0875 0.003125
Balanced Accuracy      0.7785 0.666667
```

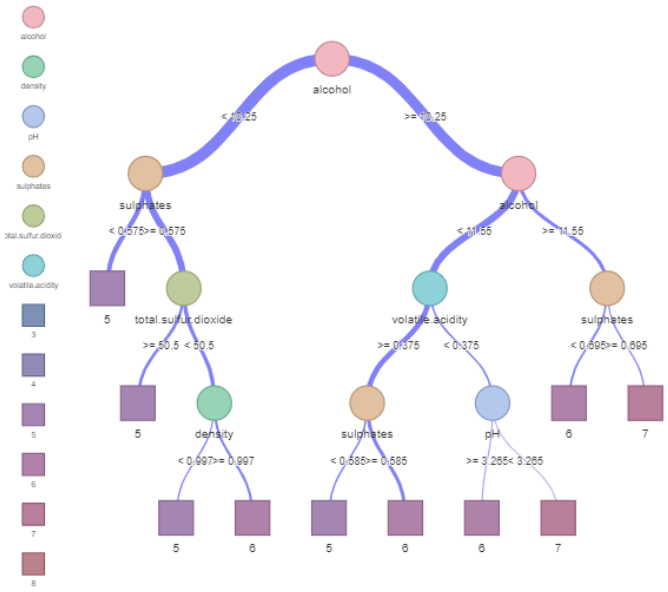
We can see that the accuracy is 69.69%. We thought the model was weak in the 3, 4, 8 range. The model simply didn't predict many 3, 4, or 8 and that went to negatively effect the prediction of 5, 6, 7 (predicting those values when it was 3, 4, 8). This model is ok but not perfect. It seems like it would do well predicting average wines but isn't fine tuned to outliers that are great wines or poor wines. This could be showing a weaker model or it could be showing the subjectivity of a wine taster and a wine scorer.

Results: This is an okay model. Like I said above, its aversion to predicting middle of the road scores on low or high rated wines could be showing a weaker model or it could be showing the subjectivity of a wine taster and a wine scorer. We can also retrieve variable importance from our model (see in Q3). As we predicted above, alcohol and sulfates are important but pH isn't as important as we though.

Model 2: RPart Decision Trees

Why: For similar reasons as the random forest, an RPart decision trees model is a good model for this data because we want to classify wines into their desired quality and this model is used as a classifier. It will be useful to compare an RPart decision trees model with the random forest and see which one performs better.

Construction: I used the same train/test split data as I did for the random forest. This will make it more appropriate to compare them later on. I fit an rpart model to the train data, again using all attributes. Here is what the tree looks like to the left.



As we've come to learn, alcohol is the most important variable and is appropriately the first split in the trees. At the next level is sulphates and alcohol which support our claims that those are the two most important variables. From there, the tree considers sulfur dioxide, density, volatile acidity, sulphates, and pH. An important takeaway is that this model will only classify wines as 5, 6, or 7. This reaffirms what the random forest model was doing. The model will focus on those quality groups and avoid predicting great or poor wines. This will cause issues as it can predict many wines well but can't pick a "great" wine.

Validation (Approach and results): When we fit this model to the validation data and compare, we find that this model does not perform as well as the random forest. This model will never predict 3, 4, or 8 whereas the random forest model would. Here is the confusion matrix to the left.

```
> confusionMatrix(rpart_pred,test$quality1)
Confusion Matrix and Statistics

          Reference
Prediction  3  4  5  6  7  8
         3  0  0  0  0  0  0
         4  0  0  0  0  0  0
         5  3  9 90 48  6  0
         6  1  2 34 73 17  1
         7  0  0  1 14 19  2
         8  0  0  0  0  0  0

Overall Statistics

               Accuracy : 0.5688
                 95% CI : (0.5125, 0.6237)
    No Information Rate : 0.4219
    P-Value [Acc > NIR] : 8.847e-08

                  Kappa : 0.3112

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 5
Sensitivity            0.0000  0.00000   0.7200
Specificity            1.0000  1.00000   0.6615
Pos Pred Value            NaN      NaN   0.5769
Neg Pred Value         0.9875  0.96562   0.7866
Prevalence             0.0125  0.03438   0.3906
Detection Rate         0.0000  0.00000   0.2812
Detection Prevalence   0.0000  0.00000   0.4875
Balanced Accuracy      0.5000  0.50000   0.6908
                     Class: 6 Class: 7 Class: 8
Sensitivity            0.5407  0.45238 0.000000
Specificity            0.7027  0.93885 1.000000
Pos Pred Value         0.5703  0.52778      NaN
Neg Pred Value         0.6771  0.91901 0.990625
Prevalence             0.4219  0.13125 0.009375
Detection Rate         0.2281  0.05937 0.000000
Detection Prevalence   0.4000  0.11250 0.000000
Balanced Accuracy      0.6217  0.69561 0.500000
```

It had an accuracy of 56.88% which is quite lower than the random forest model. While the model did select qualities at the 5, 6, and 7 levels well, the wines that took on qualities at the 3, 4, and 8 level were obviously all classified wrong.

Results: This model is not bad but it is not good. The random forest is superior.

Model 3: Generalized Linear Model Regression

Why: A glm regression model is good in this case because we have a lot of numeric data that we want to classify into quality groups. We can use quality as a numeric attribute for this analysis.

Construction: I constructed a glm regression model with the same data as the past two models, i.e. the train test split. This will make them more comparable. Again, I used all attributes for this model. Here is a summary:

```
Call:
glm(formula = quality ~ . - quality1, data = wine)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.68911  -0.36652  -0.04699   0.45202   2.02498

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.197e+01  2.119e+01   1.036   0.3002
fixed.acidity         2.499e-02  2.595e-02   0.963   0.3357
volatile.acidity     -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
citric.acid          -1.826e-01  1.472e-01  -1.240   0.2150
residual.sugar        1.633e-02  1.500e-02   1.089   0.2765
chlorides            -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
free.sulfur.dioxide   4.361e-03  2.171e-03   2.009   0.0447 *
total.sulfur.dioxide -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
density              -1.788e+01  2.163e+01  -0.827   0.4086
pH                   -4.137e-01  1.916e-01  -2.159   0.0310 *
sulphates             9.163e-01  1.143e-01   8.014 2.13e-15 ***
alcohol               2.762e-01  2.648e-02  10.429  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.4199185)

    Null deviance: 1042.17  on 1598  degrees of freedom
Residual deviance:  666.41  on 1587  degrees of freedom
AIC: 3164.3

Number of Fisher Scoring iterations: 2
```

We can see that the variables with the highest significance level, *** or 0, are volatile acidity, chlorides, total sulfur dioxide, sulphates, and alcohol. At less significance, * or 0.01, free sulfur dioxide and pH are valuable. This is useful in giving us insight into the weights of each attribute and they reaffirm some of the ideas we've gained above.

Validation (Approach and results): I fit my glm model to the test data to validate using MSE or mean squared error. I got an MSE of 0.426. Our quality for this analysis was as a numeric type so a MSE of 0.426 means that we were less than half away on predictions from the actual score. This is overall not too bad of a model and more useful if we think of quality as numeric and a rating being able to be a decimal. This would be more valuable to a numerically minded. Person.

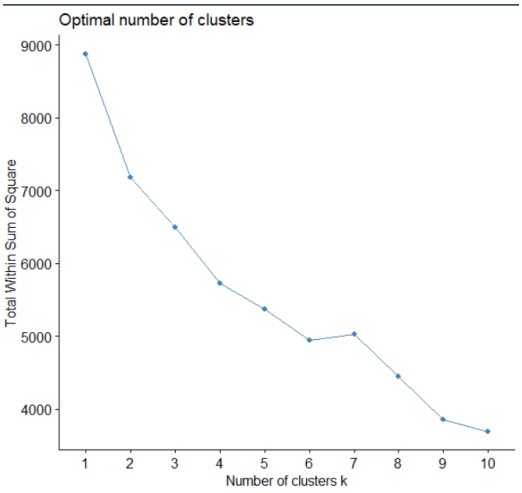
Results: Good model for treating quality as numeric and not as a factor. Gives us more insight into the meaningful variables.

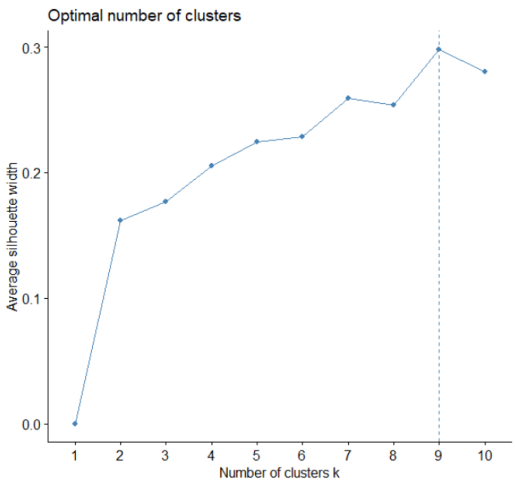
Absentee

Model 1: KMeans Clustering

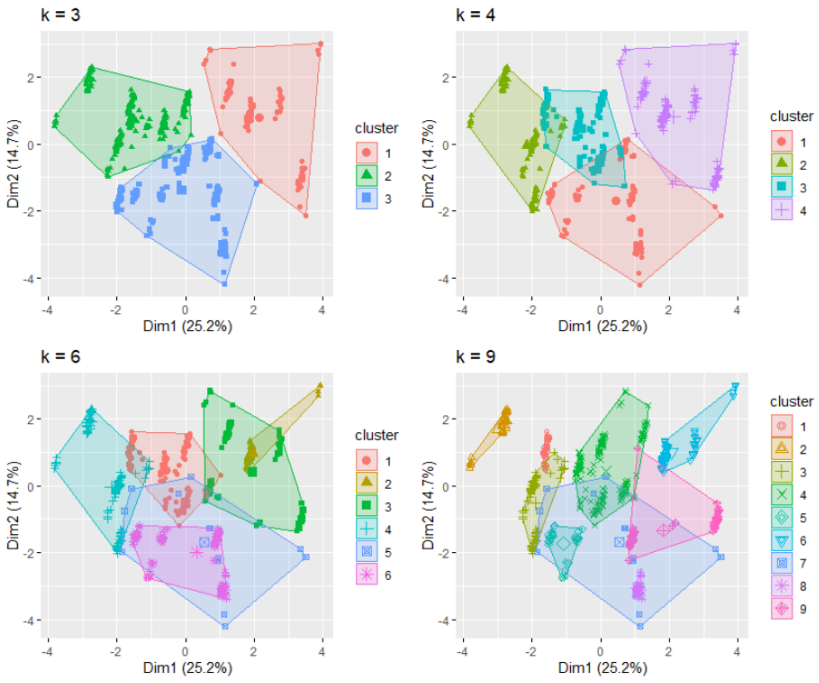
Why: Clustering on this data could be an insightful way of understand clusters of employees that miss the most work. We can then dive into attributes of each cluster to understand why certain groups are more likely to have missed more work.

Construction: First, I split the data in training data and test data using 80% as the split ratio. Then I scaled the data so that we could have generality across the attributes. Before I made the model, I ran an elbow analysis, a silhouette analysis to gain some insight into the ideal number of clusters.





It looks like 6 and 9 could be good numbers but I also ran 3 and 4 to keep things simple. Here are the clusters:



It looks like 3 or 4 clusters gives us some good insight.

Here are the defining characteristics of the clusters if we select a 3 cluster plots:

	Transportation.expense	Distance.from.Residence.to.Work	Service.time	Age
1	-0.5224653	0.5362547	1.0393907	0.88436686
2	-0.3988696	-0.8513353	-0.2023747	-0.06117828
3	0.7755781	0.4092675	-0.5908230	-0.60883532

	Work.load.Average.day	Hit.target	Son	Pet	Weight	Height
1	-0.19328056	-0.04250862	-0.4193638	-0.3229012	1.0187724	-0.2917146
2	0.03543895	0.13338831	-0.3729883	-0.4985871	-0.2900119	0.5625048
3	0.11196421	-0.09544899	0.6730362	0.7203255	-0.4914799	-0.3176577

	Body.mass.index	Absenteeism.time.in.hours
1	1.2273772	-0.14835516
2	-0.5531537	0.08931201
3	-0.3973135	0.02655945

Cluster 2 which visually stands out the most has major difference in their distance from work (small distance) and in their height (tend to be taller).

Results: Given how specific these attributes are, we may be overfit to this companies absentee data. I do not see much of a correlation between height and why someone would be absent but we see it as a defining characteristic here.

Model 2: RPart Tree

Why: At this point in order to create a classification problem I have grouped the absentee time in hours into three distinct groups: low, medium, and high. Low is under 4 hours, medium is under 8 and over 4, and high is over 8. Now that this is a classification problem, we can use decision trees to classify our data.

Construction: I used an RPart model to construct a tree diagram to classify the absent level based on all other attributes. Here is what my tree looks like:

At the first three levels of our tree, we're looking at the reason for absence which makes a lot of sense. After that, we look at if the person was a social drinker. This follows the month of absence, the height, day of the week, and month again. Toward the bottom of the tree when we start looking at the time and height, I fear we may be running into overfitting but perhaps I'm wrong. That's just my intuiton.

Validation (Approach and results): Next we take our model that was fit on the training data, create a prediction, and compare with our validation data. Here is the corresponding confusion matrix:

```
> confusionMatrix(rpart_pred,test$absentlevel)
Confusion Matrix and Statistics

          Reference
Prediction high low medium
    high     10   2      4
    low       4  70     13
    medium    6  13     26

Overall Statistics

               Accuracy : 0.7162
                 95% CI : (0.6364, 0.7872)
    No Information Rate : 0.5743
    P-Value [Acc > NIR] : 0.000254

                  Kappa : 0.4927

 Mcnemar's Test P-Value : 0.785126

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.50000     0.8235        0.6047
Specificity              0.95312     0.7302        0.8190
Pos Pred Value           0.62500     0.8046        0.5778
Neg Pred Value           0.92424     0.7541        0.8350
Prevalence               0.13514     0.5743        0.2905
Detection Rate           0.06757     0.4730        0.1757
Detection Prevalence     0.10811     0.5878        0.3041
Balanced Accuracy        0.72656     0.7768        0.7118
```

The accuracy is pretty good at 71.62%. The most incorrect predictions are at the low and medium levels. Overall this is a pretty good model that could be finetuned to be even better but its sufficient as is.

Results: A good, reliable model that could be finetuned but works well and is not overfit. It could predict high categories better but most people absent are out for an unexpected reason that would be hard to predict. Maybe we could put even more weight on the reason for absence and give the model more exposure to those reasons.

Model 3: Random Forest

Why: Random forest is a good classifier. We got good results from a basic RPart model and in the wine case we got better results from random forest than RPart so maybe a random forest model here could give us even better results than our previous model.

Construction: I constructed a random forest model with 250 trees on the same train data as the RPart model so that we can compare results. I am predicting the absent level using all other attributes. Here is the base models confusion matrix:

```
Call:
 randomForest(formula = absentlevel ~ ., data = train, ntree = 250,
nce = T)
               Type of random forest: classification
                     Number of trees: 250
No. of variables tried at each split: 4

        OOB estimate of  error rate: 23.48%
Confusion matrix:
       high low medium class.error
high     39  15     33  0.55172414
low       2 301     29  0.09337349
medium    9  51    113  0.34682081
```

Just like above, it does the best with the low and medium levels of data but not so much with the high levels. This is expected.

Validation (Approach and results): I now applied this model to the validation data and ran a confusion matrix to see the results:

```
Confusion Matrix and Statistics

          Reference
Prediction high low medium
    high     10   0      4
    low       2  77     16
    medium    8   8     23

Overall Statistics

               Accuracy : 0.7432
                 95% CI : (0.665, 0.8115)
    No Information Rate : 0.5743
    P-Value [Acc > NIR] : 1.444e-05

                  Kappa : 0.5263

 Mcnemar's Test P-Value : 0.1116

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.50000     0.9059        0.5349
Specificity              0.96875     0.7143        0.8476
Pos Pred Value           0.71429     0.8105        0.5897
Neg Pred Value           0.92537     0.8491        0.8165
Prevalence               0.13514     0.5743        0.2905
Detection Rate           0.06757     0.5203        0.1554
Detection Prevalence     0.09459     0.6419        0.2635
Balanced Accuracy        0.73438     0.8101        0.6913
```

This model does perform better than the RPart model though only slightly. It has an accuracy of 74.32%. It gets the same percentage of high predictions correct (10/20) but does better in the other areas. This goes to show that the improvements were made in the already good low and medium levels and there could not be improvements in predicting high levels. It is a tough issue to predict.

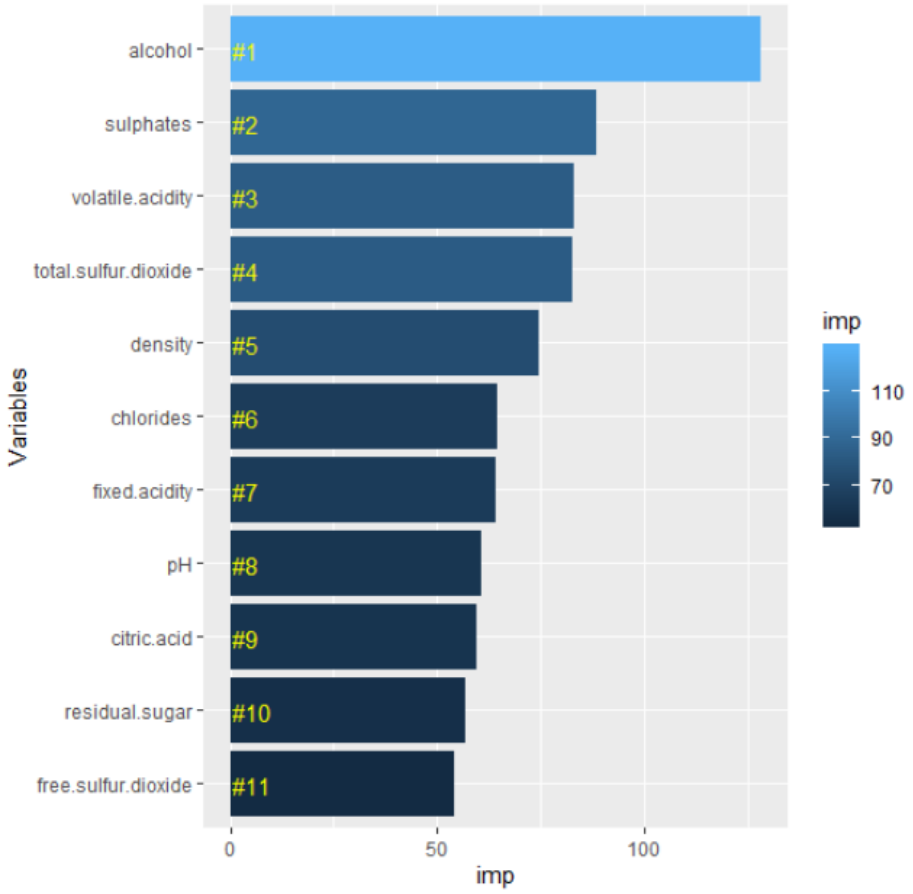
Results: High levels of absentees are likely unforeseen medical issues and cannot easily be predicted by current attributes. Because of this, they cannot be modeled easily.

3. Decisions (3%) Describe your conclusions in regard to the model fit, predictions and how well (or not) it could be used for decisions and why. Min. 1 page of text + graphics.

Wine

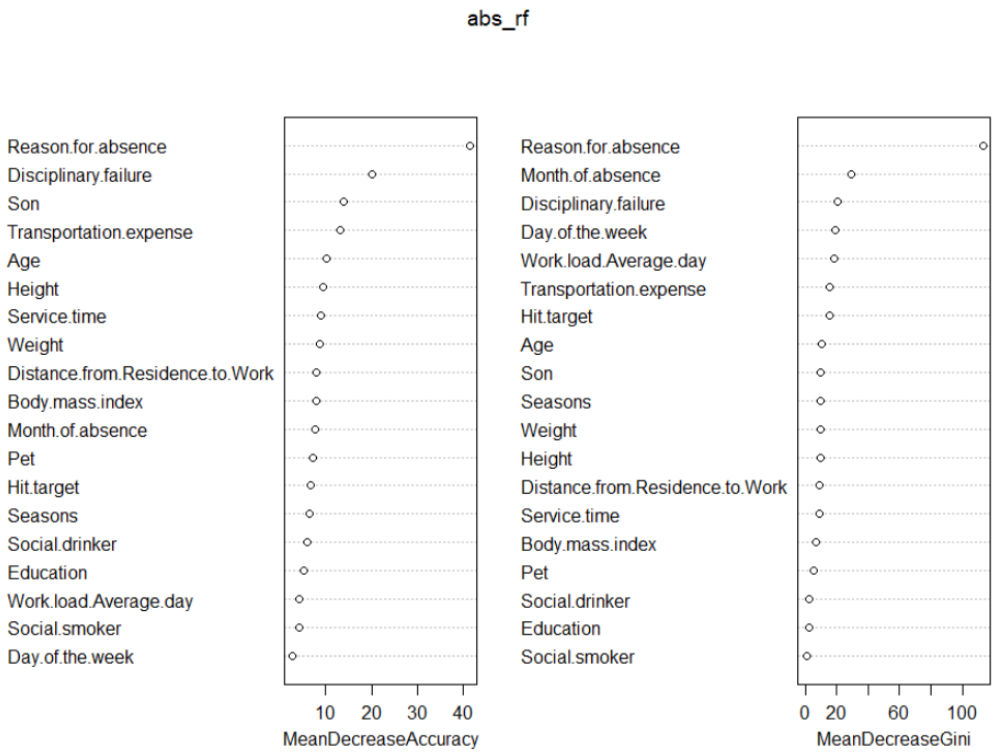
Our best model gives an accuracy of about 70%. How sufficient that is depends on the problem we're applying it to. Being apply to predict the output correctly 70% of the time is great for me in my apartment trying to predict how good a wine is but it isn't sufficient for a winery putting millions of dollars into product development. Our random forest is the best performing and outshadowing the RPart model so we can even disregard that.

We can take away the importance of certain variables on the outcomes like the alcohol levels and sulphate levels. They tend to be the most important to wine quality.



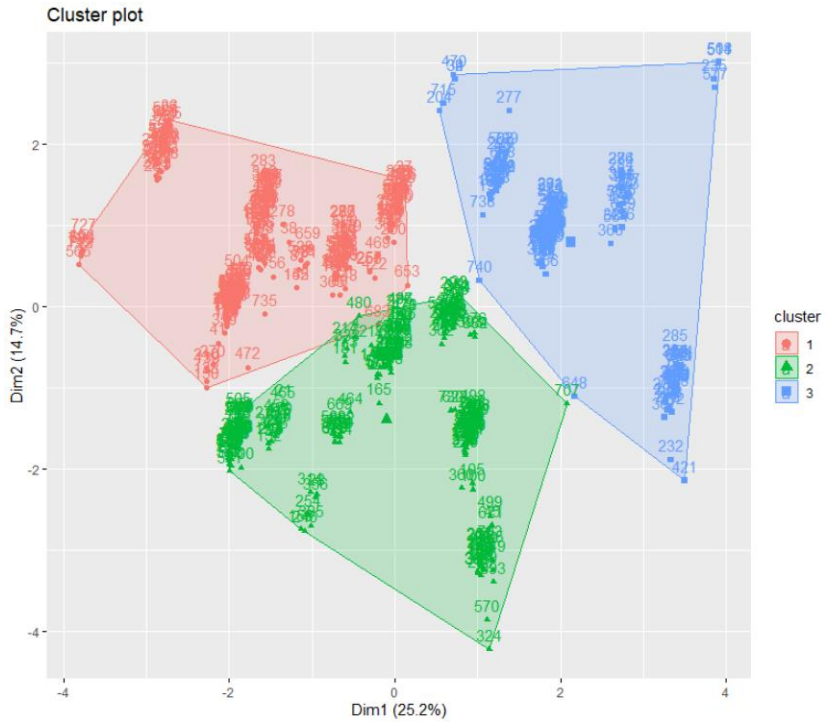
Absentee

It is really hard to predict when an employee will have a lengthy amount of time absent. This is likely due to the severity and unforeseen nature of it. Our models are pretty good at detecting low to medium absences (under 8hrs) but it's not great at above that. This model could be important to a manager trying to understand absence behavior and my random forest model would be beneficial with a good accuracy rate (74%). Variable importance will always be informative too as it can tell us what the most important predictors are to a worker's absence.



Here we can see variable importance from the random forest model. Reason for absence is the highest important variable by quite a margin.

I also think our cluster model is good for understand where an employee fits within the organization of a business.



APPENDIX – Code (https://github.com/scoutog/DataAnalyticsSpring2020**)**

```
# Read in data
absentee <- read.csv("Absenteeism_at_work.csv")
wine <- read.csv("winequality-red.csv", sep=";")
library(tidyverse)
library(reshape)
library(corrplot)
library(reshape2)
library(dplyr)
library(randomForest)
library(rpart)
library(visNetwork)
library(sparkline)
library(caret)
# 1. EDA (3%)

##############   wine
nrow(wine)
dim(wine)
table(is.na(wine))
summary(wine)
heatmap(cor(wine), Rowv = NA, Colv = NA)
corrplot(cor(wine))
wine$quality1 <- as.factor(wine$quality)

ggplot(wine, aes(x=quality, fill=quality1)) +
  geom_bar(stat="count") +
  scale_x_continuous(breaks=seq(3,8,1)) +
  ggtitle("Distribution of Quality") +
  theme(legend.position = "none")

ggplot(wine, aes(alcohol, quality,color=quality1))+
  geom_point()

ggplot(wine,aes(x=fixed.acidity,fill=quality1))+
  geom_density(alpha=0.25) +
  ggtitle("Fixed Acidity Level Distribution by Quality")

ggplot(wine,aes(x=alcohol,fill=quality1))+
  geom_density(alpha=0.25) +
  ggtitle("Alcohol Distribution by Quality")

ggplot(wine,aes(x=sulphates,fill=quality1))+
  geom_density(alpha=0.25) +
  ggtitle("Sulphate Distribution by Quality")

ggplot(wine,aes(x=pH,fill=quality1))+
```

```
  geom_density(alpha=0.25) +
  ggtitle("pH Distribution by Quality")

################# absentee
colnames(absentee)
cols <- c(2:5,12:17)
absentee[cols] <- lapply(absentee[cols], factor)
sapply(absentee, class)

col_num <- c(6:11,14,17:21)
absent_num <- absentee[,col_num]
absent_num$Work.load.Average.day <- as.numeric(absent_num$Work.load.Average.day)
absent_num$Son <- as.numeric(absent_num$Son)
absent_num$Pet <- as.numeric(absent_num$Pet)

sapply(absentee, class)

heatmap(cor(absent_num), Rowv = NA, Colv = NA)
absentee <- absentee %>%
  mutate(Month.of.absence= fct_recode(Month.of.absence,
                                      'None'='0','Jan'='1',
                                      'Feb'='2','Mar'='3',
                                      'Apr'='4','May'='5',
                                      'Jun'='6','Jul'='7',
                                      'Aug'='8','Sep'='9',
                                      'Oct'='10','Nov'='11',
                                      'Dec'='12') )

absentee <- absentee %>%
  mutate(Seasons= fct_recode(Seasons,'summer'='1','autumn'='2',
                             'winter'='3','spring'='4'))

absentee <- absentee %>%
  mutate(Education = fct_recode(Education,'highschool'='1',
                                'graduate'='2','postgraduate'='3',
                                'master& doctrate'='4'))

absentee <- absentee %>%
  mutate(Day.of.the.week = fct_recode(Day.of.the.week,"Monday"="2",
                                      "Tuesday"="3","Wednesday"="4",
                                      "Thursday"="5","Friday"="6"))

ggplot(absentee,aes(Reason.for.absence, fill= Reason.for.absence)) +
  geom_bar(stat = 'count') + coord_flip() +
  theme(legend.position='none') + xlab('Reason for absence')

ggplot(absentee, aes(Son, fill = Son)) + geom_bar()
```

```
ggplot(absentee, aes(Social.smoker, fill =  Social.drinker)) + geom_bar()

ggplot(absentee, aes(Day.of.the.week, fill =  Day.of.the.week)) + geom_bar()

ggplot(absentee, aes(Seasons,fill = Seasons)) + geom_bar()

ggplot(absentee, aes(Age, Absenteeism.time.in.hours)) +
  geom_point(aes(color=Education)) +
  geom_smooth(color="blue")

ggplot(absentee, aes(Body.mass.index, Absenteeism.time.in.hours)) +
  geom_point(aes(color=Social.smoker)) +
  geom_smooth(color="blue")

ggplot(absentee, aes(Education,fill = Education)) + geom_bar()

boxplot(absentee$Absenteeism.time.in.hours,main="Absentee Time in Hours")
hist(absentee$Absenteeism.time.in.hours, breaks=15,col="maroon")
# 2. Model Development, Validation, Optimization and Tuning (14%) Choose
# three different models
set.seed(1)
smp_size <- floor(0.8 * nrow(wine))
train_ind <- sample(seq_len(nrow(wine)), size = smp_size)
train <- wine[train_ind, ]
test <- wine[-train_ind, ]
######## Wine
# Model 1 - Random Forest
wine_rf <- randomForest(quality1 ~ . - quality, train, ntree=150)
wine_rf

rf_pred <- predict(wine_rf, test[,!colnames(test) %in% c("quality1")])
confusionMatrix(rf_pred, test$quality1)

imp <- importance(wine_rf)
varImp <- data.frame(Variables = row.names(imp),
                     Importance = round(imp[ ,'MeanDecreaseGini'],2))
rankImp <- varImp %>%
  mutate(Rank = paste0('#',dense_rank(desc(imp))))
ggplot(rankImp, aes(reorder(Variables, imp), imp,
                    fill = imp)) +
  geom_bar(stat='identity') +
  geom_text(aes(Variables, 0.5, label = Rank),
            hjust=0, vjust=0.55, size = 4, colour = 'yellow') +
  labs(x = 'Variables') +
  coord_flip()

#Model 2 - RPart Decision Trees
wine_rpart <- rpart(quality1~.-quality, train)
```

```
visTree(wine_rpart)

rpart_pred <- predict(wine_rpart, test[,!colnames(test) %in% c("quality1")], type='class')
rpart_pred
confusionMatrix(rpart_pred,test$quality1)

#Model 3 - Logistic Regression
wine_glm = glm(quality ~ .-quality1, data=train)
summary(wine_lm)

wine_glm_pred <- predict(wine_glm, test)
MSE.lm <- sum((wine_glm_pred - test$quality)^2)/nrow(test)
MSE.lm

varImp(wine_glm)
##################### Absentee
absentee <- absentee[!(absentee$Reason.for.absence==0),]
set.seed(1)
smp_size <- floor(0.8 * nrow(absentee))
train_ind <- sample(seq_len(nrow(absentee)), size = smp_size)
train <- absentee[train_ind, ]
test <- absentee[-train_ind, ]

# Model 1 - KMeans Cluster
library(cluster)
library(factoextra)
library(gridExtra)
absent_num_sc <- scale(absent_num)
fviz_nbclust(absent_num_sc, kmeans, method = "wss")
fviz_nbclust(absent_num_sc, kmeans, method = "silhouette")
gap_stat <- clusGap(absent_num_sc, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
fviz_gap_stat(gap_stat)

k3 <- kmeans(absent_num_sc, centers = 3, nstart = 25)
fviz_cluster(k3, data = absent_num_sc)

k4 <- kmeans(absent_num_sc, centers = 4, nstart = 25)
k6 <- kmeans(absent_num_sc, centers = 6, nstart = 25)
k9 <- kmeans(absent_num_sc, centers = 9, nstart = 25)

p1 <- fviz_cluster(k3, geom = "point", data = absent_num_sc) + ggtitle("k = 3")
p2 <- fviz_cluster(k4, geom = "point",  data = absent_num_sc) + ggtitle("k = 4")
p3 <- fviz_cluster(k6, geom = "point",  data = absent_num_sc) + ggtitle("k = 6")
p4 <- fviz_cluster(k9, geom = "point",  data = absent_num_sc) + ggtitle("k = 9")
grid.arrange(p1, p2, p3, p4, nrow = 2)
best <- kmeans(absent_num_sc, centers=3, nstart=25)
best$center
```

```
fviz_cluster(best, absent_num_sc)
# Model 2 - Tree
temp <- as.integer(as.character(absentee$Absenteeism.time.in.hours))
for (i in 1:length(temp)) {
  if(temp[i] >= 1 & temp[i] <=4){
    absentee$absentlevel[i] = "low"
  } else if(temp[i] > 4 & temp[i] <= 8){
    absentee$absentlevel[i] = "medium"
  } else { absentee$absentlevel[i] = "high"}
}
table(absentee$absentlevel)
absentee$absentlevel <- as.factor(absentee$absentlevel)
absentee$Work.load.Average.day <- as.numeric(absentee$Work.load.Average.day)
rm <- c(1,21)
absentee <- absentee[,-rm]
smp_size <- floor(0.8 * nrow(absentee))
train_ind <- sample(seq_len(nrow(absentee)), size = smp_size)
train <- absentee[train_ind, ]
test <- absentee[-train_ind, ]
library(tree)
absent_tree <- rpart(absentlevel~., train)
visTree(absent_tree)

rpart_pred <- predict(absent_tree, test[,!colnames(test) %in% c("absentlevel")], type='class')
rpart_pred
confusionMatrix(rpart_pred,test$absentlevel)

# Model 3 - Random Forest
abs_rf <- randomForest(absentlevel~., train, ntree=250, importance=T)
abs_rf

abs_rf.pred <- predict(abs_rf, test, type = "class")
confusionMatrix(abs_rf.pred, test$absentlevel)

importance(abs_rf)
varImpPlot(abs_rf)

# 3. Decisions (3%) Describe your conclusions in regard to the model fit,
# predictions and how well (or not) it could be used for decisions and why.
# Min. 1 page of text + graphics.

################## wine
################## absentee

## See document
```