Katie Baerveldt

MK 6460

Spring B 2020

Assignment 2

Part I: Using Logistic Regression

This part is a "structured" portion that requires you to complete a series of tasks following each step.

A telecom service firm collected 4832 customers' information and also it recorded whether the customers have churned in the previous year. Please use this dataset ("ConsumerChurn_csv_1.csv") and use logistic regression to understand the drivers behind customer churn.

Nearly all my variables were categorical to begin with, so I transformed them into factored dummies to fit into a regression formula correctly.

There are two numerical variables, MonthlyCharges and TotalCharges, and one integer, tenure. My target variable is Churn, which is already in a binary format necessary for regression.

```
'data.frame':    4832 obs. of  19 variables:
 $ gender          : Factor w/ 2 levels "Female","Mal
 $ SeniorCitizen   : Factor w/ 2 levels "No","Yes": 1
 $ Partner         : Factor w/ 2 levels "No","Yes": 2
 $ Dependents      : Factor w/ 2 levels "No","Yes": 2
 $ tenure          : int  9 9 4 13 3 9 71 63 7 65 ...
 $ MultipleLines   : Factor w/ 2 levels "No","Yes": 1
 $ InternetService : Factor w/ 2 levels "DSL","Fiber
 $ OnlineSecurity  : Factor w/ 2 levels "No","Yes": 1
 $ OnlineBackup    : Factor w/ 2 levels "No","Yes": 2
 $ DeviceProtection: Factor w/ 2 levels "No","Yes": 1
 $ TechSupport     : Factor w/ 2 levels "No","Yes": 2
 $ StreamingTV     : Factor w/ 2 levels "No","Yes": 2
 $ StreamingMovies : Factor w/ 2 levels "No","Yes": 1
 $ Contract        : Factor w/ 3 levels "Month-to-mon
 $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2
 $ PaymentMethod   : Factor w/ 4 levels "Bank transfe
 $ MonthlyCharges  : num  65.6 59.9 73.9 98 83.9 ...
 $ TotalCharges    : num  593 542 281 1238 267 ...
 $ Churn           : Factor w/ 2 levels "No","Yes": 1
```

The first model I ran without doing a train/test split provided me with an informative but messy initial view of the relationships in the data. I removed the scientific notation of the logits for easier understanding:

(Intercept)	genderMale
4.3846995	0.9565158
SeniorCitizenYes	PartnerYes
1.2082978	0.9544797
DependentsYes	tenure
0.9600641	0.9261440
MultipleLinesYes	InternetServiceFiber optic
1.6581708	6.0618095
OnlineSecurityYes	OnlineBackupYes
0.8172470	1.0072117
DeviceProtectionYes	TechSupportYes
1.1333253	0.8622228
StreamingTVYes	StreamingMoviesYes
1.7387501	1.7462748
ContractOne year	ContractTwo year
0.6008079	0.2955885
PaperlessBillingYes	PaymentMethodCredit card (automatic)
1.3756813	1.0305437
PaymentMethodElectronic check	PaymentMethodMailed check
1.4172770	1.0189096
MonthlyCharges	TotalCharges
0.9586334	1.0004788

This shows my variables in terms of likelihoods in relation to churn occurring. For example, having multiple lines slightly increases the likelihood of churn by 1.66 times.

My train/test split approach broke my training set into 3,387 observations and my test data into 1,445 observations. I then created the following fits:

- fit; only the demographic variables including gender, SeniorCitizen, Partner, Dependents, tenure
  - AIC 3650.4
  - ~72% accuracy
  - Without releveling the variables, I see here that men have very little effect on the Churn outcome
  - Having a partner and length of tenure also hold little predictive properties, but their logits are still higher than the value assigned to men
- fit2; only the service type related model including MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod
  - AIC 3434.9
  - ~75.5% accuracy
  - This model is a little more complex and has more Fisher scoring iterations than the last
  - Error evaluated through AIC is slightly lower than the last model
  - The variables that seem to have the least predictive properties include having multiple lines and using a credit card

    - This is surprising, because in the original model I ran without the test/train split it seemed to suggest that having multiple lines had a stronger correlation with Churn
- fit3; all variables
    - AIC 3266.9
    - ~78%
    - My risk decreases when I include all variables in my training fit
    - However, this may be too complex, it's hard to see any real relationship between the target and the other variables, but we know they exist

It's still hard to come to a decision on which of my models is the best to move forward with or use as a basis for further decision-making toward model tuning. I ran an Anova analysis on all three of my methods for further input on how they stack up:

```
Analysis of Deviance Table

Model 1: Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure +
    MultipleLines + InternetService + OnlineSecurity + OnlineBackup +
    DeviceProtection + TechSupport + StreamingTV + StreamingMovies +
    Contract + PaperlessBilling + PaymentMethod + MonthlyCharges +
    TotalCharges
Model 2: Churn ~ MultipleLines + InternetService + OnlineSecurity + OnlineBackup +
    DeviceProtection + TechSupport + StreamingTV + StreamingMovies +
    Contract + PaperlessBilling + PaymentMethod
Model 3: Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure
  Resid. Df Resid. Dev Df Deviance              Pr(>Chi)
1      3365     3222.9
2      3372     3404.9 -7  -181.99 < 0.00000000000000022 ***
3      3381     3638.4 -9  -233.50 < 0.00000000000000022 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Even though my full variable model had a lower error score and slightly higher overall accuracy, this analysis shows that there are imperfections in the algorithm of fit3 as seen by the more significant Chi statistics on the first and second fit models.

This tells me that there are predictive properties to be explored further via some variables, and that some of the correct noise needs to be removed.

First fit accuracy: 0.716263 Second fit accuracy: 0.7550173 Third fit accuracy: 0.782699

Though my all-variable third fit had highest accuracy, the differential in scores isn't enough to convince me that this is the model I will use, especially since the anova() function has essentially shown me that more tuning is needed with variable selection.

Interpret the coefficients and generate conclusions that may be useful for marketing/business purposes

Can check VIFs
multi-collinearity

```
                                  Estimate Std. Error z value
(Intercept)                       0.3910535  1.8158896   0.215
genderMale                       -0.0139959  0.0864601  -0.162
SeniorCitizenYes                  0.1439781  0.1062843   1.355
PartnerYes                        0.0137703  0.1021147   0.135
DependentsYes                    -0.1381021  0.1208008  -1.143
tenure                           -0.1080527  0.0125830  -8.587
MultipleLinesYes                  0.3587048  0.2200788   1.630
InternetServiceFiber optic        0.9696293  1.0030576   0.967
OnlineSecurityYes                -0.3386506  0.2258958  -1.499
OnlineBackupYes                  -0.1775800  0.2237619  -0.794
DeviceProtectionYes              -0.0542082  0.2217323  -0.244
TechSupportYes                   -0.2633352  0.2283725  -1.153
StreamingTVYes                    0.1772929  0.4082477   0.434
StreamingMoviesYes                0.2551604  0.4100952   0.622
ContractOne year                 -0.4866296  0.1461492  -3.330
ContractTwo year                 -1.4223411  0.2500272  -5.689
PaperlessBillingYes               0.2701945  0.1020555   2.648
PaymentMethodCredit card (automatic) -0.0838244  0.1512622  -0.554
PaymentMethodElectronic check     0.4292813  0.1229138   3.493
PaymentMethodMailed check         0.0782929  0.1622993   0.482
MonthlyCharges                   -0.0142466  0.0399551  -0.357
TotalCharges                      0.0008244  0.0001342   6.143
                                            Pr(>|z|)
(Intercept)                                 0.829494
genderMale                                  0.871403
SeniorCitizenYes                            0.175529
PartnerYes                                  0.892729
DependentsYes                               0.252947
tenure                          < 0.0000000000000002 ***
MultipleLinesYes                            0.103124
InternetServiceFiber optic                  0.333707
OnlineSecurityYes                           0.133836
OnlineBackupYes                             0.427422
DeviceProtectionYes                         0.806862
TechSupportYes                              0.248871
StreamingTVYes                              0.664087
StreamingMoviesYes                          0.533812
ContractOne year                            0.000869 ***
ContractTwo year                      0.000000012798 ***
PaperlessBillingYes                         0.008108 **
PaymentMethodCredit card (automatic)        0.579465
PaymentMethodElectronic check               0.000478 ***
PaymentMethodMailed check                   0.629523
MonthlyCharges                              0.721417
TotalCharges                          0.000000000808 ***
---
```

Altogether it's more difficult to discern the predictive power of any variable here via the intercept coefficient alone.

However, under Pr(>|z|) we find that the function has denoted significant values automatically

For marketing/business purposes, I would first make an algorithm that only includes tenure, ContractOne year, ContractTwo year, PaperlessBillingYes , PaymentMethodElectronic check, and TotalCharges since these appear to hold significance in predicting Churn.

I would make another algorithm that seeks to just remove values that consistently show little to know predictive power, such as: genderMale, PartnerYes, DeviceProtectionYes, PaymentMethodCredit card (automatic), PaymentMethodMailed check, and MonthlyCharges.Should I continue to find issues, I will add variables back as necessary.

Finally, I would relevel some of the variables here that caught my attention to see if the other level has significance.

I chose to relevel my gender and Partner categorical variables to see if it produced useful insights outside of my existing models and kept the rest of the variables included. My releveled fit produced an AIC of 3,266.9, and an accuracy score of ~78%. This matched the score I got for fit3, so this didn't tell me anything new. Moving forward I would likely remove more variables and research a better way to incorporate releveling in a regression.

I finally attempted a stepwise regression model, however this produced an AIC score of 3,259.9 and an accuracy of 71.6%.

Either this means that this isn't the best approach moving forward, or it's telling me that my fit isn't ready to be processed by a cleaner method

Moving forward, I would explore further fits and different train/test splits alongside each other, as there appears to be relationships between the target and some variables that come to light once some of the noise is removed

For example, some variables such as OnlineSecurityYes did not score a significant coefficient until some of the less useful variables we identified were removed. In fit2, OnlineSecurityYes shows more significance than when lost among the all-variable fit.

OnlineSecurityYes -0.52066 0.10430 -4.992 0.0000005974952284 *** OnlineSecurityYes -0.3386506 0.2258958 -1.499 0.133836

When releveling we usually want to check those variables that have 3 or more levels. We try to reset the base level.

Apply exp() and discuss more about the findings. (coefficients and significance)

Part II: Freestyle

This portion is a "freestyle" task. To complete this part, you will first search and decide a dataset that is appropriate for logistic regression.

Then, you will use this dataset and adopt logistic regressions to explore rich insights from this dataset.

Description according to source: https://www.kaggle.com/abhisheikreddy646/house-prediction-for-zipcode#80111.csv

The Zestimate was created to give consumers as much information as possible about homes and the housing market, marking the first time consumers had access to this type of home value information at no cost.

"Zestimates" are estimated home values based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property.

The context of this dataset is meant to perform housing price prediction based on city zipcode.

This dataset includes 66 observations of housing prices spanning several decades in the same zipcode. The original variables include ID, price, sqft_living, bedrooms, bathrooms, year_built, zipcode, address, Latitude, and Longitude. For the purpose of this task I removed ID, zipcode, Latitude and Longitude since many of these variables did not pertain to the task at hand and/or only had one factor level.

This dataset is little bit too small for running regression

A preliminary analysis of the target variable, price, shows me the following:

- Mean: $804,979.70
- Median: $766,534.50
- Min: $182,500.00
- Max: $1,910,991.00

It's easy to see that housing prices are trending upward, however there are outliers that skew the data. I chose the median as a point at which to dummy my target variable into a binary format easier for predictions. Moving forward, all housing prices under the median were factored as "Less expensive" and anything over the median was "More expensive."

Out of my 66 observations, 48 properties fell under the median while 18 properties ran over the median.

I thought this dataset was interesting because it could either be used for a regression incorporating GIS and/or time series. I try to take these elements into account later by factoring year_built and address further.

A quick summary of my data prepped for regression analysis:

```
  sqft_living       bedrooms      bathrooms      year_built
 Min.   :  658   Min.   :1.000   Min.   :1.000   2019   :13
 1st Qu.: 2404   1st Qu.:2.250   1st Qu.:3.000   2018   : 5
 Median : 3036   Median :4.000   Median :3.000   1978   : 4
 Mean   : 3623   Mean   :3.636   Mean   :3.248   1981   : 3
 3rd Qu.: 3934   3rd Qu.:5.000   3rd Qu.:4.000   1982   : 3
 Max.   :35719   Max.   :6.000   Max.   :7.000   1989   : 3
                                                 (Other):35
                          address       Latitude       Longitude
 Centennial, CO 80111          :23   Min.   :39.62   Min.   :-104.9
 Cherry Hills Village, CO 80111: 1   1st Qu.:39.62   1st Qu.:-104.9
 Englewood, CO 80111           :18   Median :39.62   Median :-104.9
 Greenwood Village, CO 80111   :24   Mean   :39.62   Mean   :-104.9
                                     3rd Qu.:39.62   3rd Qu.:-104.9
                                     Max.   :39.62   Max.   :-104.9
```

I can see that there are some outliers already, particularly with a couple of very expensive properties being present in the data due to the max values seen in square footage.

The price mean shows me this same finding because most of the properties fall below the median threshold I used to divvy up the category.

I can also see that most houses in this area are new and were built just last year in 2019.

This summary allows me to see that latitude and longitude won't have any significance in my regression because there's only one value presented for each variable - these may have been included for approaches involving geocoding.

I made the decision to randomize my data before performing regression since I don't have many observations to begin with.:

```
   sqft_living bedrooms bathrooms year_built                    address
28        1541        3         3       2016        Centennial, CO 80111
22        4341        3         3       1997 Greenwood Village, CO 80111
37        1626        3         3       2002 Greenwood Village, CO 80111
44        2988        4         3       2019        Centennial, CO 80111
47         891        1         1       1989 Greenwood Village, CO 80111
9         3715        4         3       1979 Greenwood Village, CO 80111
            price
28 Less expensive
22 Less expensive
37 Less expensive
44 Less expensive
47 Less expensive
9  More expensive
```

I used a traditional 70/30 split to build my models. My training set had 47 observations while my test set had 19 observations. After an initial run of my first fit model, I decided to factor my year_built model into “Pre-90s” vs. “Post-90s” due to the noise the previous 32 levels of year_built factors were in my data. My models were produced as follows:

- **fit_all**
  - After simplifying the factor levels in year_built, my first algorithm was able to pick up on variables that have significance in the predictive power toward price, which I used to build my next fit
  - I’m surprised that sqft_livnig doesn’t have more significance in predicting how expensive the price will be, in fact it seems to have almost no significance when I include all variables
  - Furthermore, it doesn’t seem like the neighborhood influences pricing as much as I thought it would

```
Call:
glm(formula = price ~ ., family = binomial, data = trainset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7326  -0.7471  -0.3732   0.7153   2.4350

Coefficients:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -3.342e+00  1.872e+00  -1.785   0.0743 .
sqft_living                        -2.507e-06  1.212e-04  -0.021   0.9835
bedrooms                            8.884e-01  4.422e-01   2.009   0.0446 *
bathrooms                          -2.438e-01  4.941e-01  -0.493   0.6218
year_builtPre-90s                  -2.711e+00  1.153e+00  -2.351   0.0187 *
addressCherry Hills Village, CO 80111 -1.390e+01  2.400e+03  -0.006   0.9954
addressEnglewood, CO 80111          1.606e+00  1.097e+00   1.464   0.1432
addressGreenwood Village, CO 80111  3.278e-01  9.890e-01   0.331   0.7403
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 57.251  on 46  degrees of freedom
Residual deviance: 42.965  on 39  degrees of freedom
AIC: 58.965

Number of Fisher Scoring iterations: 15
```

- **fit_significant**
  - My coefficients are much more easily interpretable, and my error went down even lower
  - So far this tells me that houses built before the 90s are less likely to have more expensive pricing in this zip code
  - Since I’m going to evaluate my models altogether using anova() and other comparisons, I’ll forego doing a confusion matrix and mean accuracy on every model - I don’t want to rely on mean accuracy alone when evaluating which model I’ll choose

```
glm(formula = price ~ bedrooms + year_built, family = binomial,
    data = trainset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6991  -0.7325  -0.4916   0.8671   2.0847

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.2804     1.4374  -2.282   0.0225 *
bedrooms            0.7424     0.3423   2.169   0.0301 *
year_builtPre-90s  -1.7415     0.8798  -1.979   0.0478 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 57.251  on 46  degrees of freedom
Residual deviance: 45.992  on 44  degrees of freedom
AIC: 51.992

Number of Fisher Scoring iterations: 5
```

- **fit_relevel**
  - I releveled year_built to reflect Post-90s, which predictably showed that this heavily influences higher pricing on houses
  - However, I could derive that from my last fit
  - I added address back in and got a slightly higher error score, but I can see now that the Englewood neighborhood has predictive properties regarding higher pricing

```
glm(formula = price ~ bedrooms + year_built + address, family = binomial,
    data = trainset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5771  -0.7829  -0.4019   0.7007   2.4584

Coefficients:
                                     Estimate Std. Error z value Pr(>|z|)
(Intercept)                           -6.3567     2.1874  -2.906  0.00366 **
bedrooms                               0.7762     0.3678   2.110  0.03482 *
year_builtPost-90s                     2.6030     1.1359   2.292  0.02193 *
addressCherry Hills Village, CO 80111 -14.3648  2399.5449  -0.006  0.99522
addressEnglewood, CO 80111             1.5655     1.0925   1.433  0.15188
addressGreenwood Village, CO 80111     0.2801     0.9720   0.288  0.77326
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 57.251  on 46  degrees of freedom
Residual deviance: 43.257  on 41  degrees of freedom
AIC: 55.257

Number of Fisher Scoring iterations: 15
```

I chose to evaluate all of my models together using Anova as well as Adjust R Squared.

Anova:

```
Analysis of Deviance Table

Model 1: price ~ sqft_living + bedrooms + bathrooms + year_built + address
Model 2: price ~ bedrooms + year_built
Model 3: price ~ bedrooms + year_built + address
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        39     42.965
2        44     45.992 -5  -3.0271   0.6958
3        41     43.257  3   2.7349   0.4343
```

Adjusted R-squared:

- fit_all: -0.02994432
- fit_significant: 0.09185053
- fit_relevel: 0.03481947

The first analysis of all three of my models using the anova() method doesn't produce any highly significant chi-squared values, but I can see that my third model using three variables has the most significant chi-squared value. This model has the least likelihood of the observations being due to chance out of all models observed.

My second analysis involving adjusted R-squared shows me that the explanatory power of the model increases when I limit my model to the two significant variables of # bedrooms and year_built. However, the quality of the model decreases in terms of adjusted R-squared when I add address, even though it allowed for me to gather more surface-level insights. The adjusted R-squared will decrease when the new term doesn't improve the model by a sufficient amount, so while this third model may have produced better initial results it will require more tuning if I want to analyze address.

Generate implications for the context

This data would be extremely helpful for any buyers or sellers in real estate in this area of Colorado. Buyers can enter negotiations with a better knowledge base, and sellers can interpret the qualities of their property (# bedrooms, year built) in order to gauge what price they can get. Overall, anyone in the real estate market would be able to use an algorithm such as this to strategize how to competitively price their property or provide competitive offers.

The data has some limitations but the analysis is good. Have more interpretations

55+35=90 Nice job!