BMI Study using the NHanes 2013-2014 Dataset

Thomas Arnold, 2021 IBM Machine Learning Class Project

The following analysis was conducted to determine whether age, gender, marital status, and caloric intake on a sample day interact to affect a person's body mass index (BMI). The data was taken from the National Health and Nutrition Examination Survey (NHanes) data collected from 2013-2014. Six different regression analyses were attempted.

1) Linear regression with y = BMI, 9 Features
2) Linear regression with y = log(BMI) without polynomial features, 9 Features
3) Linear regression with y = log(BMI) with polynomial features, 39 Features
4) Linear regression with y = log(BMI) with polynomial features and standardized scaling
5) Lasso regression with y = log(BMI) with polynomial features and standardized scaling
6) Ridge regression with y = log(BMI) with polynomial features and standardized scaling

After examination of the results from the six models, it appeared that linear regression with log(BMI) of the outcome and polynomial features produced the highest R Squared for both the train and test samples. Standardization of the polynomial features had no effect on the accuracy of the linear regression model.

Note that there did not appear to be any overfitting with the linear model. The R squared for the test sample using linear regression was as high as for the train sample. Lasso and ridge regression produced the same results as those seen with straight linear regression. This suggests that there is no overfitting on the linear regression model.

Main objective of the analysis

There were two main objectives, 1) Interpretation and 2) Model evaluation.

My first goal was interpretation. I was interested in determining which of the following factors affect BMI. These factors were the following.

1. Age
2. Gender
3. Marital status
4. Daily caloric intake

I also wanted to determine if interaction effects were present between these four factors.

My second goal was to determine which regression method works best. I wanted to determine whether I had a problem with overfitting that can occur with simple linear regression. Therefore, in addition to linear regression, I tried lasso and ridge regression.

I probably need to do more with this dataset. Even though the R Squared is high, many of the terms were nonsignificant at a p<.05 level. The respondents with unknown marital status were significantly different than those with known values. Perhaps some imputation methods might be used.

Brief description of the data set

I used the National Health and Nutrition Examination Survey (NHanes) data collected from 2013-2014.

See https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2013

There are many hundreds of variables in many different types of tables available in the NHanes data. The NHanes data is provided as a series of tables stored in a SAS (.xpt) format. I used the following three data tables in these analyses. The variables used are listed as well.

1. Dependent variable
   - a. BMI data (BMX_H.XPT)
     - i. Original name - BMXBMI
       - 1. Type – Float
     - ii. Recoded initially as y
     - iii. Recoded again and transformed to ylog
       - 1. Used np.log1p to transform
2. Independent variables
   - a. Demographics data (DEMO_H.XPT)
     - i. Age
       - 1. Original name – RIDAGEYR
         - a. Type – Float
       - 2. Recoded to – Age
         - a. Type – Float
     - ii. Gender
       - 1. Original name – RIAGENDR
         - a. Type – Categorical
         - b. Values (1,2)
       - 2. Recoded to – Male
         - a. Type – Dichotomous
         - b. Values (1,0)
     - iii. Marital status
       - 1. Original name – DMDMARTL
         - a. Type – Categorical
         - b. Numeric codes
       - 2. Recoded to MaritalStatus
         - a. Coded using descriptions as text
         - b. NaN coded to Unkown
         - c. Type Categorical

3. Marital status was recoded to dummies
    a. The Unknown column was used as the omitted variable
    b. Type Dichotomous dummies
    c. Value Counts

MaritalStatus	**N**
Unknown	2,980
Married	2,616
Never married	955
Divorced	579
Living with partner	374
Widowed	336
Separated	149

b. Daily diet data (DR1TOT_H.xpt)
    i. Original name - DR1TKCAL
        1. Type – Float
        2. Description - Total caloric intake on first day of interviews
    ii. Recoded and transformed to CalIntlog
        1. Transformed using np.log1p of caloric intake

Brief summary of data exploration, data cleaning, and feature engineering

The following steps were used in the data exploration process. The three raw dataframes (demographics, BMI, and diet) were first examined using 1) shape(), 2) head(), and 3) describe. Features were checked for skew and value counts. Histograms were created for continuous variables such as BMI and caloric intake. Columns were renamed to provide better understanding. Rows with NaN in the BMI and caloric intake columns were dropped. The skewed variables (BMI and caloric intake) were transformed using the np.log1p function. A set of y and X dataframes was created. The X dataframe was reduced to four features. Dummy variables were created from the MaritalStatus column. Polynomial terms and interaction terms were added. The features were scaled to a standard scale with mean zero and standard deviations as the values.

1. Dataframe (df) Steps
    a. df.shape
    b. df.head()
    c. df.describe()
    d. print(column names in loop)
2. Column (column) Steps
    a. Data exploration
        i. column.skew()
        ii. column.value_counts()
        iii. Plotted histograms
    b. Data cleaning
        i. Renamed columns

        - 1. df.rename(columns={“Old Name” : “New Name”})
    - ii. Dropped NaN rows
        - 1. df.dropna(subset=[“Column Name”], inplace=True)
    - iii. Converted Marital Status Codes to MaritalStatus text
        - 1. Created a MaritalStatus swap dataframe
        - 2. Merged the MaritalStatus swap with main dataframe
- c. Feature engineering
    - i. Created new y dataframe from BMI
        - 1. y = BMIStudydf['BMXBMI']
        - 2. Transformed BMI (y) to log of BMI (ylog)
    - ii. Transformed CaloricIntake to log of CaloricIntake (CalIntlog)
    - iii. Created new X dataframe with four columns
        - 1. X = BMIStudydf[["Age", "Male", "MaritalStatus", "CalIntlog"]]
    - iv. Transformed MaritalStatus column to dummy variables
        - 1. Used get_dummies()
        - 2. Removed MaritalStatus_Unknown column
            - a. X = X.drop(columns="MaritalStatus_Unknown")
    - v. Used PolynomialFeatures to get interactions and squared values
    - vi. Used StandardScaler to get all variables on same scale
    - vii. Used the linear regression output to drop columns with coefficients = 0

Summary of training six linear regression models

Six regression models were tested.

1. Linear regression with y = BMI without polynomial features, 9 Features
2. Linear regression with ylog = log(BMI) without polynomial features, 9 Features
3. Linear regression with ylog = log(BMI) with polynomial features, 39 Features
4. Linear regression with ylog = log(BMI) with polynomial features and standardized scaling
5. Lasso regression with ylog = log(BMI) with polynomial features and standardized scaling
6. Ridge regression with ylog = log(BMI) with polynomial features and standardized scaling

With the linear regression, I used the sklearn LinearRegression and the statsmodel OLS packages. The statsmodel OLS package provides p values, which are helpful in interpretation. This was a little redundant, but useful from a training perspective.

I found that 6 features from the initial linear regression models created using the Xpf features predicting ylog produced coefficients = 0, so I dropped those.

With Lasso and Ridge regression, I deviated a little from the course example. I created a function for testing the Lasso and Ridge regression models using 10 different Alpha levels. This was developed by using the machine learning code described on the Analytics Vidhya web site as a model. Their code was modified slightly to provide additional information on the R Squared values for the train and test samples.

See https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/

The Lasso and Ridge regression code sets each tested ten different Alpha values. Train and test samples were created from the Xpfss (X with polynomial features and standard scaling) dataframe and the ylog dataframe. The Xpfss_train, Xpfss_test, ylog_train, and ylog_test samples were passed to the Lasso and Ridge functions along with Alpha values. The output from the Lasso and Ridge funtions consisted of the following 55 columns.

1. rss_train
2. rss_test
3. r2_train
4. r2_test
5. coef_non_zero
6. coef_sum
7. Intercept
8. Feature coefficients (55 columns)

R Squared Output from regression models

Trial #	Model	Regression	Columns	Train R^2	Test R^2
1	X,y	Linear	9	28.38%	28.90%
2	X,ylog	Linear	9	35.06%	36.32%
3	Xpf,ylog	Linear	39	45.92%	45.90%
4	Xpfss,ylog	Linear	39	45.92%	45.90%
5	Xpfss,ylog	Lasso	39	45.92%	45.90%
6	Xpfss,ylog	Ridge	39	45.92%	45.90%

Recommendations

Based on the results of the six regression models, it appeared that overfitting was not an issue. The best Lasso and Ridge model accuracy was seen with Aplha = $1x10^{-15}$, which was essentially Linear Regression. Therefore, the linear regression model is equal to Lasso and Ridge in terms of accuracy, and better in terms of interpretability. The statsmodel OLS regression provided p values, which can be used to provide an idea of which variables had the greatest impact.

The full linear model output is provided in the appendix. The most significant values are provided below. For some reason, the respondents with missing marital status data appeared to have the highest BMI. Age appeared to have a nonlinear effect, with the formula .267 * Age(ss) -.217 * $Age(ss)^2$. Being male and married also seemed to be associated with an increased BMI.

Most Significant Statsmodel Features

Feature	Coefficient	p	Lower CI	Upper CI
const	3.254	.000	3.249	3.260
Age x MaritalStatus_Unknown	.117	.000	.101	.134
Age^2	-.217	.000	-.274	-.160
Male x MaritalStatus_Married	.025	.002	.009	.040
MaritalStatus_Unknown	-.131	.004	-.220	-.041
MaritalStatus_Unknown^2	-.131	.004	-.220	-.041
Age	.267	.009	.068	.466

Key Findings and Insights

This was a useful exercise for learning about regression and machine learning. I think that I can reuse many of the code sections in other projects. I learned that Lasso and Ridge regression models are not always needed. Note that I have used Lasso regression in my work as a population health data scientist and I found that Lasso and Ridge regression helped prevent overfitting in that context.

Next Steps

In thinking about next steps regarding the BMI study, I would probably want so see what other variables might be useful. I found that the log transformation of the BMI values improved the model accuracy from 28% to 35%. Adding dummies also improved the accuracy from 35% to 46%. There was little degradation in the test sample.

Many of the variables seemed to have poor predictive power, based on the p values. I would probably remove some of them. I would be trying to see if I could find more information on the respondents who did not fill out the marital status responses. With more data, I might be able to do some imputation.

Appendix- Statsmodel output

Feature	Coefficient	p	Lower CI	Upper CI
const	3.254	.000	3.249	3.260
Age x MaritalStatus_Unknown	.117	.000	.101	.134
Age^2	-.217	.000	-.274	-.160
Male x MaritalStatus_Married	.025	.002	.009	.040
MaritalStatus_Unknown	-.131	.004	-.220	-.041
MaritalStatus_Unknown^2	-.131	.004	-.220	-.041
Age	.267	.009	.068	.466
CalIntlog x MaritalStatus_Separated	.079	.102	-.016	.174
Age x MaritalStatus_Living_with_partner	-.014	.108	-.032	.003
Age x Male	-.017	.150	-.040	.006
Age x MaritalStatus_Separated	-.016	.152	-.037	.006
MaritalStatus_Married^2	-.055	.162	-.133	.022
MaritalStatus_Married	-.055	.162	-.133	.022
MaritalStatus_Separated^2	-.031	.219	-.081	.019
MaritalStatus_Separated	-.031	.219	-.081	.019
CalIntlog x MaritalStatus_Married	.091	.235	-.059	.242
Male x MaritalStatus_Separated	-.005	.255	-.013	.003
CalIntlog x MaritalStatus_Living_with_partner	.046	.366	-.053	.144
CalIntlog x MaritalStatus_Unknown	.078	.375	-.095	.252
Male x CalIntlog	-.042	.408	-.141	.057
CalIntlog x MaritalStatus_Widowed	-.048	.417	-.163	.068
CalIntlog	-.047	.427	-.161	.068
Age x MaritalStatus_Widowed	.018	.460	-.030	.065
Male x MaritalStatus_Unknown	.006	.489	-.011	.024
Male x MaritalStatus_Widowed	.003	.497	-.005	.011
CalIntlog^2	.035	.527	-.074	.145
Age x MaritalStatus_Divorced	-.009	.548	-.039	.021
MaritalStatus_Living_with_partner	-.014	.586	-.064	.036
MaritalStatus_Living_with_partner^2	-.014	.586	-.064	.036
Male^2	.013	.611	-.038	.065
Male	.013	.611	-.038	.065
Male x MaritalStatus_Living_with_partner	-.002	.712	-.011	.008
Age x CalIntlog	.033	.719	-.146	.211
MaritalStatus_Widowed	.011	.721	-.052	.075
Male x MaritalStatus_Divorced	.001	.770	-.008	.011
MaritalStatus_Divorced^2	.007	.828	-.054	.067
MaritalStatus_Divorced	.007	.828	-.054	.067
CalIntlog x MaritalStatus_Divorced	-.010	.865	-.125	.105
Age x MaritalStatus_Married	.000	.988	-.035	.035
MaritalStatus_Widowed^2	.011	.721	-.052	.075