Final Project
On the Divorce Dataset
STSCI 4740, Fall’19

Group 10:
Zackary Downey (zjd6)
Jonathan Mac (jm2583)
Yi Zhu (yz2629)

Instructor:
Yang Ning
Date Submitted:
December 10th, 2019

Introduction

For our analysis, we used the Divorce data set found via the UCI Machine Learning repository, here: https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set#. The goal is to create a machine learning model to predict divorce and analyze the parameters chosen. In this project, we 1) predict divorce with multiple methods, 2) compare/contrast methodologies and test MSE values, 3) determine our preferred method, and 4) analyze questions associated with the predictors that best predicted divorce.

In total, we incorporated 18 unique models into our analysis. These include both regression and classification techniques. After directly comparing Expected Test MSE (ETMSE) across all models, we determined that QDA was the preferred method, with lowest ETMSE of 1.764%. Key parameters used in this model are: Atr11, Atr18, Atr19, Atr17, and Atr40.

The key parameters were crucial in measuring the strength of a marital relationship, so it is not surprising that these may strongly predict a potential divorce. We will elaborate on our methods and provide summary statistics and conclusions for some key ones.

Data at First Look

Our divorce data set contains 170 rows (observations) each containing 54 parameters based on questions posed to married couples, and response variable titled "Class", which indicates whether the couple is divorced or still married. The response values of 1 and 0 indicate a status of divorced and still married, respectively. Each of the 54 parameters is an integer value between 0 and 4, notated by "AtrX", where X is the parameter tied to the question number X. We, along with the class a as whole, theorize that these values indicate the level of agreement with the question, but converted so that a value of 0 is the most associated with remaining married while 4 is the most associated with a divorce. This meaning may differ by the question, making interpretation tricky, but is something that our models can handle.

To better understand the data set, we built a correlation matrix to determine correlation coefficients between each of the 54 parameters and response Class. This correlation matrix guides us by indicating some key parameters for our modelling. For example, parameter Atr40 has the highest correlation with Class, a value of 0.9386. The question associated with Atr40 is:

"We're just starting a discussion before I know what's going on".

This question appears slightly confusing, but our group interprets this as, "we are commonly arguing out of nowhere for no reason". A high level of agreement (i.e. Atr40 = 4) with this question appears to be a negative indication that the couple is not happy. A scatter plot of the factor levels of Atr40 against our response Class reveals that in the survey, every participant who is still married never gave a response above 2 (Figure 1).

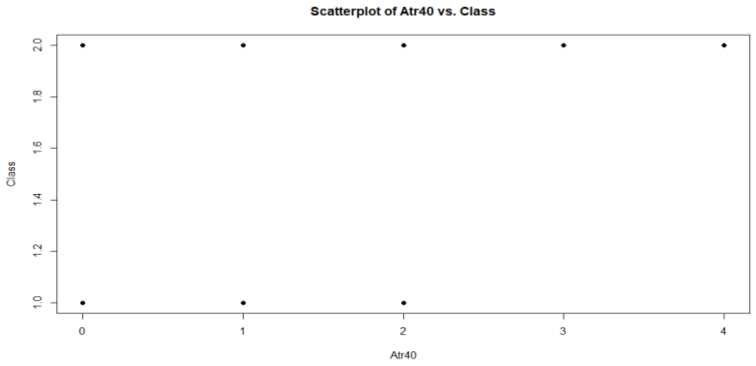


Figure 1 - Scatter plot of Atr40 vs. Class indicates that a high level of agreement with this (negative) question is associated with divorce.

We wanted to find the 5 best predictors influencing Class so we performed a 5-fold cross validation using correlation coefficients. The top 5 parameters that appeared most frequently: Atr11, Atr18, Atr19, Atr17, and the aforementioned Atr40. Agreeing in the positive way for these questions seems important for a happy marriage. These questions are respectively as follows:

1. Atr11 = I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other.
2. Atr18 = My spouse and I have similar ideas about how marriage should be.
3. Atr19 = My spouse and I have similar ideas about how roles should be in marriage.
4. Atr17 = We share the same views about being happy in our life with my spouse.
5. Atr40 = We're just starting a discussion before I know what's going on.

Creating Machine Learning Models

Our analysis begins using all 54 parameters in the most basic models. Later on, we use this information to simplify some subset selection steps where convenient R functions are not readily accessible. To be consistent, we applied the 5-fold cross validation and used the same training and testing sets on all methods. However, because of the sample() function used to generate folds for validation sets, the size of each fold is not exactly the same.

<u>Basic Model</u>

Using a 5-fold cross validation, we applied regression/classification methods to obtain the most basic models, all of which include 54 predictors. We obtained a final Test MSE for each method to measure the prediction accuracy of each model.

In our code, we provide the RSS of each *regression* method, and also provide a classification error even for regression methods. To accomplish this, we wrote a simple function to associate each regression prediction with a Class value of 0 or 1 (i.e.predict() function chooses 1 if predicts >=0.5; otherwise 0). This allows us to use the regression output to make a classification prediction. This is not the preferred method, but allows us to compare methods more directly.

For a classification problem like this, a logistic regression seems more appropriate. However, after fitting the model without subset selection or shrinkage methods, we found multiple warnings of “fitted probabilities numerically 0 or 1 occurred” and “algorithm did not converge”. The latter is due to too many parameters and too few observations, which violates the “one in ten rule”[1] of regression analysis. The number of independent predictors also exceeds the maximum input allowed by the glm function. As a fix, we simply added a control “control=list(maxit=1000)” into our R code to override the glm default. However, the first warning still occurs because of a complete or quasi-complete separation of the data; the model is extremely unstable because the MLE does not exist when the data are well-separated.

Smoothing Splines and Natural Splines, in their simplest form, are designed to model one predictor and therefore do not make much sense for all 54 predictors. We decided to fit a Generalized Additive Model (GAM) as an extension to the splines by including them as fitted functions within GAM. GAMS are applicable to classification problems, and also overcome the curse of dimensionality that this dataset suffers from.

For GAM with Smoothing Splines, in order to simplify the model, we set all degrees of freedom to 5 for the 54-predictor Smoothing Splines functions. Then we ran a 5-fold cross validation, and constructed a confusion matrix for each fold, to calculate the test error rate. We intended to apply the same procedure above for GAM with Natural Splines but ended up encountering major errors. One possible explanation is that Natural Splines are designed to fit continuous data and thus performed badly on this dataset.

The final results for all nine basic methods can be seen in Figure 2 below. We have also included one graphic for the decision tree method, shown as quite a basic tree (3 external nodes) in Figure 3.

Method (Classification)	**Expected Test MSE / Classification Error**
Linear Regression	2.277%
Logistic Regression	3.878%
Logistic Regression (with Control)	3.148%
Polynomial Linear Regression	1.764%
Polynomial Logistic Regression	1.567%
LDA	2.277%
KNN	2.277%
Decision Tree	2.883%

[1] https://en.wikipedia.org/wiki/One_in_ten_rule

GAM with Smoothing Spline	2.865%

Figure 2 - Basic models for all nine methods attempted along with Test MSE.

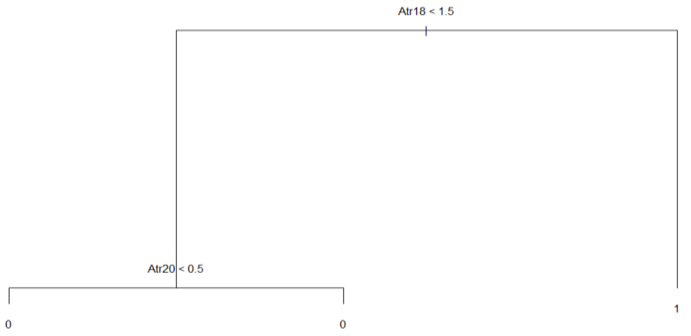


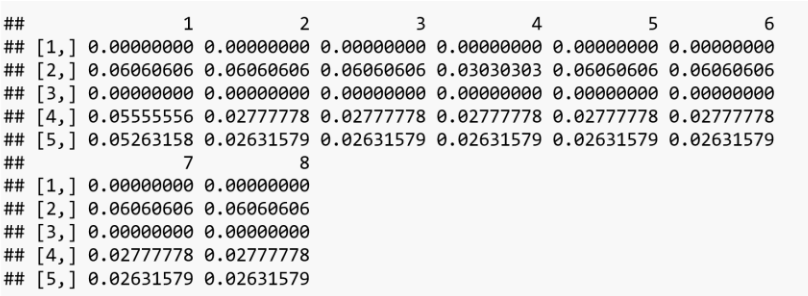
Figure 3 - One example decision tree generated from full parameter decision tree method.

Although these methods with 5-fold CV yield low classification errors, fitting 54 parameters leads to overfitting and difficult interpretation. Also, this data set contains 170 rows of data and 54 predictors. Because there is a relatively large number of predictors “p” compared to the number of observations “n”, Curse of Dimensionality implies that a flexible model would be a poor representation of its true fit. So by removing irrelevant predictors, we can obtain a model that is more easily interpretable. Thus, less flexible models are preferred here.

Subset Selection

We applied subset selection and shrinkage to obtain more sparse models. To obtain the fewest predictors possible, we preferred BIC over AIC as a criterion for subset selection. Similarly, we preferred lasso regression over ridge regression for shrinkage. We performed best/forward/backward selection on linear regression and lasso on both linear regression and logistic regression.

The limitations of best subset selection on linear regression is the computational feasibility. We ran best subsets with varying values for nvmax, which restricts the best model to have nvmax predictors. R took extremely long (over 20 minutes) and started to lag after limiting nvmax to 9 predictors. Thus, we set the maximum number of parameters to be eight. For best subset selection, we generated a matrix of test errors by running cross validation within each fold across each parameter count up to eight. The kth row represented the kth fold over which CV was run; the p-th column represents the p-th predictor model. Each value in the matrix is the test MSE. Then, we averaged each column, to obtain the average of k MSEs for each p-th predictor model. Finally, we averaged this mean vector to obtain a Final Test MSE. The matrix R outputs for this is shown in Figure 4.



```
##              1          2          3          4          5          6
## [1,] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [2,] 0.06060606 0.06060606 0.06060606 0.03030303 0.06060606 0.06060606
## [3,] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## [4,] 0.05555556 0.02777778 0.02777778 0.02777778 0.02777778 0.02777778
## [5,] 0.05263158 0.02631579 0.02631579 0.02631579 0.02631579 0.02631579
##              7          8
## [1,] 0.00000000 0.00000000
## [2,] 0.06060606 0.06060606
## [3,] 0.00000000 0.00000000
## [4,] 0.02777778 0.02777778
## [5,] 0.02631579 0.02631579
```

Figure 4 - Best subset selection (regsubsets) matrix of cross validation errors.

This provided a Final Test MSE and also coefficients for each p-th predictor model. Out of all the eight final models, R chose the fourth predictor model as the best with lowest MSE. The predictors included were Atr6, Atr18, Atr29, and Atr40. Best subset selection considers all 2^p predictor models, while forward and backward selection considers only 2p-1 predictor models. Since the latter have more computational tolerance, we can remove the nvmax constraint (or set to 54 all predictors) because of the significantly fewer total comparisons required.

First for shrinkage in each fold (i.e. lasso), we used the glmnet package to perform 10-fold cross validation to obtain lambda, the optimal value of our tuning parameter. Second, we used the optimal lambda in that fold to perform a lasso regression in that fold. Third, we would obtain an MSE and append it to a vector. Finally, we take the average of the mean vector to obtain a single value for MSE.

The results of these five additional methods can be found in Figure 5 below.

Method	**Regression Type**	**Expected Test MSE**
Best Subset Selection	Linear	2.331%
Forward Selection	Linear	2.073%
Backward Selection	Linear	2.320%
Lasso Regression	Linear	2.277%
Lasso Regression	Logistic	2.277%

Figure 5 - Subset selection methods applied to linear/logistic regression with associated test MSE.

Using Correlation Coefficients to Fit Model

Unfortunately, R does not have many straight forward functions to handle subset selection for LDA, QDA, etc. Otherwise, we'd apply similar logic to these methods to get sparser versions of each model. For this classification problem, we have utilized a less preferred but suitable alternate method of subset selection - letting our correlation analysis from earlier determine the top five parameters to use. Then, we refit some of our earlier models with these five parameters only: Atr11, Atr18, Atr19, Atr17, and Atr40. Though not flawless, this is the most we could do.

We now refit LDA, QDA, KNN, and Decision Tree with these five predictors only. We must carefully interpret this model's Test MSE, considering our method of generating this subset is not perfect by the book. In a best case scenario, we would use a refined subset selection technique, however due to the consistency of these top five parameters in our looped correlation analysis, we believe this is suitable for a basic subset selection. The results of these updated models can be seen in Figure 6 below.

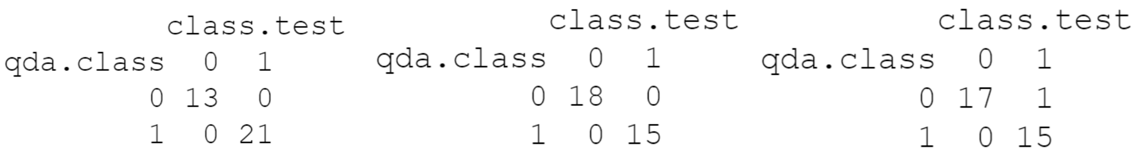
Method	**Expected Test MSE**
LDA	2.277%
QDA	1.764%
KNN	32.439%
Decision Tree	3.030%

Figure 6 - Additional models from top-five parameters chosen from correlation analysis.

Model Comparison Analysis

Now, we compared the results from 18 different unique models, to determine our preferred one. Overall, almost all models had a very similar test MSE, in the range of ~1.5% up to ~3.0%. KNN with five parameters only was an outlier with ETMSE of 32.439%, but the 19 others fell within a two percentage point range on classification error. The top three models strictly in terms of lowest Test MSE were QDA (1.764% with five parameters), polynomial linear regression (1.764% with all predictors), and polynomial logistic regression (1.567% with all predictors).

Since this is classification, we prefer QDA or polynomial logistic regression. The latter may overfit the model with too many parameters, while the former uses an imperfect (correlation matrix) way to choose the top five parameters. Although with a higher test MSE, Lasso with logistic regression is classification based and also limits the number of parameters in a statistically sound way. It also generates sparse models, unlike polynomial logistic regression. This helps since we have few observations relative to our high dimension in this dataset. Though all three methods are suitable, QDA with top five parameters is our best final model. Three of the 5-fold confusion matrices can be seen in Figure 7 below. To better summarize all models, we have provided a graphical representation of their ETMSE values in Figure 8 below.



```
           class.test              class.test              class.test
qda.class   0  1        qda.class   0  1        qda.class   0  1
        0  13  0                0  18  0                0  17  1
        1   0 21                1   0 15                1   0 15
```

Figure 7 - Confusion matrices from QDA run with only top five parameters from correlation analysis.

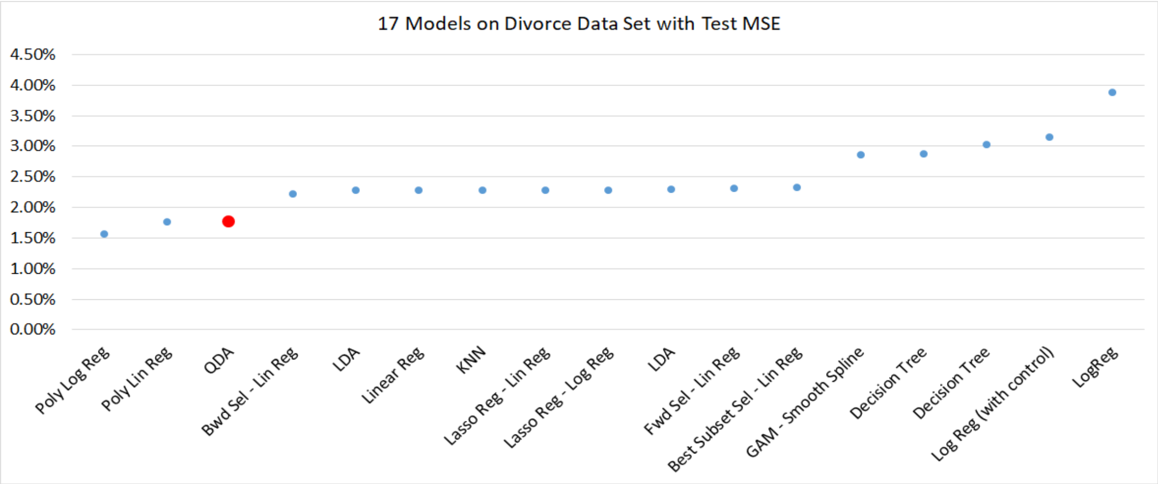


Figure 8 - Each model and test MSE. Note, we have left out KNN with 5-parameters - it is a major outlier and we wish to show the rest of the model types.

Real-Life Application

In order to apply our final model, one of our group members and his significant other chose to take the survey and run the QDA model chosen against the results to determine divorce or not. This is a slightly more whimsical application of these model results, but the group is happy to report that the predicted output was Class = 0 for all 5-folds in the loop run on this new testing set. To confirm this prediction, the updated (five-parameter) decision tree was also run on this data, providing Class = 0 as well. (Hooray!)

Conclusion

We created and analyzed twenty unique model types on the divorce data set predicting class divorce or not. Our final model of choice QDA, with five specific parameters (Atr11, Atr17, Atr18, Atr19, Atr40), performed well with a Test MSE / classification error of 1.764%. As opposed to LDA, QDA is more flexible with non-linear boundaries which may lead to a low bias and high variance.

Though all relevant, some questions appeared more frequently throughout subset analysis, especially Atr18 and Atr40. Both of these parameters showed up in our 5-fold correlation analysis, as well as in best subset selection. Atr6, Atr17, and Atr26 also appeared in subset selection, although not in the correlation analysis. All key questions are summarized below in Figure 9.

Parameter / Attribute	Where it Appeared	Question
Atr11	Correlation	I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other.
Atr18	Corr / Subset	My spouse and I have similar ideas about how marriage should be.
Atr19	Correlation	My spouse and I have similar ideas about how roles should be in marriage.
Atr17	Correlation	We share the same views about being happy in our life with my spouse.
Atr40	Corr / Subset	We're just starting a discussion before I know what's going on.
Atr6	Subset	We don't have time at home as partners.
Atr26	Subset	I know my spouse’s basic anxieties.

Figure 9 - Summary of key questions/attributes that appeared in our analysis.

These questions appear to be quite important for a couple, as they strongly indicate not only happiness in the marriage, but so ability to spend quality time together, confidence in knowing their partner, and a uniform set of ideals about marriage. Disagreeing with these concepts could lead to insecurity, confusion, or even mistrust, and eventually a separation.

Overall, the scope of our statistical analysis has been comprehensive and mostly all-encompassing. On the positive side, we used many different methods, considered gains and drawbacks, and directly compared model types. However, taking the shotgun approach limits us from deeply analyzing and refining one specific model type to lower its MSE as much as possible.

Although we have chosen QDA as our “final” model, we feel strongly that other models attempted in this analysis are equally as valid. These include lasso for logistic regression, decision trees, LDA, and many more. Since the Estimated Test MSEs are relatively close for quite a small data set of observations, we believe a different set of folds or including more data could have changed the results considerably. In brief, this data set was manageable and interesting to interpret with our subset-selected models. We feel confident that our model would be successful on a larger version of the same data set.