CS4342-HW3

Ivan Martinovic

November 2021

1 Conceptual and Theoretical Questions

1. Using a little bit of algebra, prove that (4.2) is equivalent to (4.3) - check below equations. In other words, the logistic function representation and logit representation for the logistic model are equivalent.

$$P(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \tag{4.2}$$

$$\frac{P(X)}{1-P(X)} = e^{\beta_0+\beta_1 X} \tag{4.3}$$

We start by simplifying $1 - P(X)$:

$$1 - P(X) = 1 - \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{\beta_0+\beta_1 X}}$$

Next we divide (4.2) by $1 - P(X)$:

$$\frac{P(X)}{1-P(X)} = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} * (1 + e^{\beta_0+\beta_1 X}) = e^{\beta_0+\beta_1 X} = (4.3)$$

q.e.d.

2. It was stated in the text that classifying an observation to the class for which (4.12) is largest is equivalent to classifying an observation to the class for which (4.13) is largest. Prove that this is the case. In other words, under the assumption that the observations in the kth class are drawn from a N(μ_k, σ^2) distribution, the Bayes' classifier assigns an observation to the class for which the discriminant function is maximized.

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{-1}{2\sigma^2}(x-\mu_k)^2)}{\sum_{l=1}^{K} \pi_l \frac{-1}{\sqrt{2\pi}\sigma} exp(\frac{1}{2\sigma^2}(x-\mu_l)^2)} \tag{4.12}$$

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k) \tag{4.13}$$

Lets start by analyzing (4.12) and noticing that the denominator is constant for all values of k. Since it is a sum, let's denote it with an S.

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} exp(\frac{-1}{2\sigma^2}(x-\mu_k)^2)}{S}$$

Next lets, for easier writing, lets replace $\frac{1}{\sqrt{2\pi}\sigma}$ with A:

$$p_k(x) = \frac{\pi_k A * exp(\frac{-1}{2\sigma^2}(x-\mu_k)^2)}{S}$$

Next lets do the log trick (since whenever a function reaches its maximum, its log also reaches its maximum).

$$log(p_k(x)) = log(\pi_k) + log(A) - \frac{x^2}{2\sigma^2} + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} - log(S)$$

Now notice that if we were to take the first derivative of the above function with respect to k to find the maximum, we would find that the terms which do not depend on k (namely $log(A)$, $\frac{x^2}{2\sigma^2}$ and $log(S)$) would have a derivative equal to 0. Therefore when it comes to the first derivative with respect to k the above expression is equivalent to:

$$log(\pi_k) + x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} = \delta_k(x) \ ... \ (4.13)$$

Hence when finding k which maximizes 4.12, we are also maximizing the discriminant 4.13

q.e.d.

3. This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class specific mean vector and a class specific covariance matrix. We consider the simple case where p = 1; i.e. there is only one feature. Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, $X\ N(\mu_k, \sigma_k^2)$. Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.
Hint: For this problem, you should follow the arguments laid out in Section 4.4.2, but without making the assumption that $\sigma_1^2 = \sigma_2^2 = ... = \sigma_k^2$

We start with the equation for $p_k(x)$:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} exp(\frac{-1}{2\sigma_k^2}(x-\mu_k)^2)}{\sum_{l=1}^{K} \pi_l \frac{-1}{\sqrt{2\pi}\sigma_l} exp(\frac{1}{2\sigma_l^2}(x-\mu_l)^2)}$$

Similar to the last question, let's replace the denominator with S and $\frac{1}{\sqrt{2\pi}}$ with B:

$$p_k(x) = \frac{\pi_k B * \frac{1}{\sigma_k} exp(\frac{-1}{2\sigma_k^2}(x-\mu_k)^2)}{S}$$

Next we take the log of the above expression:

$$log(p_k(x)) = log(\pi_k) + log(B) - log(\sigma_k) - \frac{x^2}{2\sigma_k^2} + x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} - log(S)$$

The Bayes' classifier assigns the class k, for which the above equation is maximized. Notice how this time when we take the derivative with respect to k of the above equation, (because B and S are constants) it will be equivalent to taking the derivative of the equation:

$$\delta_k(x) = log(\pi_k) - log(\sigma_k) + x^2 \frac{1}{2\sigma_k^2} + x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2}$$

Notice how now the quadratic term of x is present, and this is the same reason why the decision boundary is quadratic. To find the decision boundary between to classes k and l, we find those values of x for which $\delta_k(x) = \delta_l(x)$, which comes down to solving the quadratic equation:

$$-x^2(\frac{1}{2\sigma_k^2} - \frac{1}{2\sigma_l^2}) + x(\frac{\mu_k}{\sigma_k^2} - \frac{\mu_l}{\sigma_l^2}) - (\frac{\mu_k^2}{2\sigma_k^2} - \frac{\mu_l^2}{2\sigma_l^2}) + (log(\pi_k) - log(\pi_l)) - (log(\sigma_k) - log(\sigma_l)) = 0$$

, which is a quadratic equation of x.

q.e.d.

4. We now examine the differences between LDA and QDA.

(a) If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

We always expect the more flexible model to perform better on a training set. In this case the more flexible model is QDA. However, on the actual test set, we expect the model which better reflects the relationship to have better performance. In this case that model is LDA since it gives linear decision boundaries.

(b) If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

In this case we expect QDA to perform better for both the test and training set. We expect it to perform better on the training set because it is more flexible than LDA; and we expect it to perform better on the test set because it better fits the Bayes decision boundary.

(c) In general, as the sample size n increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?

This again depends on the shape of the true Bayes' decision boundary. However, in general we expect the prediction accuracy of QDA to improve relative to LDA, as the sample size increases.
This is because LDA has to estimate only one covariance matrix, which, When there are p predictors, requires estimating $p(p+1)/2$ parameters.
QDA estimates a separate covariance matrix for each class, for a total of $Kp(p+1)/2$ parameters, which is quite a lot more.
Here is where the bias-variance trade-off comes into play. LDA is a much less flexible classifier than QDA, and so has substantially lower variance. This can

potentially lead to improved prediction performance. But there is a trade-off: if LDA's assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.
Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial. In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable. However, assuming that the shape is non-linear we still expect LDA to outperform QDA for very small sample sizes.

(d) True or False: Even if the Bayes decision boundary for a given problem is linear, we will probably achieve a superior test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.

This statement is false. It is generally true that the model which best reflects the Bayes' decision boundary will have a lower test error error rate. In this case that model is LDA. QDA, because of its much higher flexibility, will start overfitting the values for the training data set. We can also see empirical proof for this in Scenarios 1-3 in section 4.5. In all three scenarios the decision boundary was linear, which lead to LDA outperforming QDA when it comes to the test error rate.

5) Suppose we collect data for a group of students in a statistics class with variables X1 =hours studied, X2 =undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, $\beta_0 = -6$, $\beta_1 = 0.05$, $\beta_2 = 1$.

(a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

$p(A) = \frac{exp(\beta_0+\beta_1 X_1+\beta_2 X_2)}{1+exp(\beta_0+\beta_1 X_1+\beta_2 X_2)} = \frac{exp(-6+0.05*40+1*3.5)}{1+exp(-6+0.05*40+1*3.5)} = \frac{exp(-0.5)}{1+exp(-0.5)} = 0.37754$

(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

$$\frac{exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}{1 + exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)} = 0.5$$
$$\frac{exp(-6 + 0.05 * h + 1 * 3.5)}{1 + exp(-6 + 0.05 * h + 1 * 3.5)} = 0.5$$

$$\frac{exp(0.05 * h - 2.5)}{1 + exp(0.05 * h - 2.5)} = 0.5$$

$$exp(0.05 * h - 2.5) = 0.5 + 0.5 * exp(0.05 * h - 2.5)$$

$$0.5 * exp(0.05 * h - 2.5) = 0.5$$

$$exp(0.05 * h - 2.5) = 1$$

$$exp(0.05 * h - 2.5) = exp(0)$$

$$0.05 * h - 2.5 = 0$$

$$0.05 * h = 2.5$$

$$h = 50$$

Answer: He would have to study for 50 hours.

6. Suppose that we wish to predict whether a given stock will issue a dividend this year ("Yes" or "No") based on X, last year's percent profit.We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was $\bar{X} = 10$, while the mean for those that didn't was $\bar{X} = 0$. In addition, the variance of X for these two sets of companies was $\sigma^2 = 36$. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was X = 4 last year.
Hint: You will need to use Bayes' theorem.

Label stocks which issue dividens with 1, and those who do not with 0.
Using Bayes' theorem we have:

$$p_k(x) = P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_k(x)}$$

From the problem text we have the following info:

$$\pi_0 = 0.2$$

$$\pi_1 = 0.8$$

$$f_0(x) = \frac{1}{\sqrt{72\pi}} exp(\frac{-1}{72} x^2)$$

$$f_1(x) = \frac{1}{\sqrt{72\pi}} exp(\frac{-1}{72} (x - 10)^2)$$

Using a calculator we find that:

$$f_0(4) = 0.0532413343$$

$$f_1(4) = 0.0403284541$$

First let's calculate the denominator.

$$\sum_{l=1}^{K} \pi_l f_k(x) = 0.2 * 0.05324 + 0.8 * 0.040328 = 0.0429110302$$

Next we can calculate $p_0(x)$ and $p_1(x)$:

$$p_1(x) = \frac{0.8 * 0.040328}{0.850977} = 0.7518524526$$

The probability that a company will issue a dividend given that its percentage profit is 4 is 0.7518524526.

7. Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get an error rate of 20% on the training data and 30% on the test data. Next we use 1-nearest neighbors (i.e. K = 1) and get an average error rate (averaged over both test and training data sets) of 18%. Based on these results, which method should we prefer to use for classification of new observations? Why?

We choose the logistic regression classification procedure. This all comes down to which procedure gives a lower error rate for the test data set. The catch of this question is that for the KNN procedure we are given the combined error rate for both the training and test data sets. However, if we remember how KNN works, we will find that the error rate of a KNN method on a training set is exactly 0%. If K = 1, and we are using our training data set, then the single nearest neighbor to each point (which is specified by a set of predictors) is that point itself, so we are assigning the correct class to it. This then implies that the test error rate must be 36% (because $((36\% + 0\%)/2 = 18\%)$, which is worse than the logistic regression's test error rate of 30%.

2 Applied Questions

1. This question should be answered using the Weekly data set.

(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

Note: Encode 'Direction' with 'Up' = 1 and 'Down' = 0
Means of each predictor:

```
Year          2000.048669
Lag1             0.150585
Lag2             0.151079
Lag3             0.147205
Lag4             0.145818
Lag5             0.139893
Volume           1.574618
Today            0.149899
Direction        0.555556
dtype: float64
```

We see that the means of the "Lag" predictors are close in value, although they seem to decrease going from Lag1 to Lag5.
The mean of the direction is 0.5555, which is higher than 0.5, implying that on average the market has a positive return.

Covariance of the data set:

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
Year	36.399284	-0.459163	-0.474864	-0.427333	-0.443261	-0.434777	8.567416	-0.461572	-0.066585
Lag1	-0.459163	5.555508	-0.415889	0.326233	-0.396511	-0.045544	-0.258209	-0.416825	-0.058592
Lag2	-0.474864	-0.415889	5.556647	-0.421334	0.324822	-0.403543	-0.339986	0.328723	0.085190
Lag3	-0.427333	0.326233	-0.421334	5.571970	-0.420064	0.338092	-0.275856	-0.396366	-0.026888
Lag4	-0.443261	-0.396511	0.324822	-0.420064	5.570916	-0.421759	-0.243134	-0.043535	-0.024112
Lag5	-0.434777	-0.045544	-0.403543	0.338092	-0.421759	5.575665	-0.233053	0.061290	-0.021327
Volume	8.567416	-0.258209	-0.339986	-0.275856	-0.243134	-0.233053	2.844742	-0.131493	-0.015089
Today	-0.461572	-0.416825	0.328723	-0.396366	-0.043535	0.061290	-0.131493	5.555107	0.843656
Direction	-0.066585	-0.058592	0.085190	-0.026888	-0.024112	-0.021327	-0.015089	0.843656	0.247141

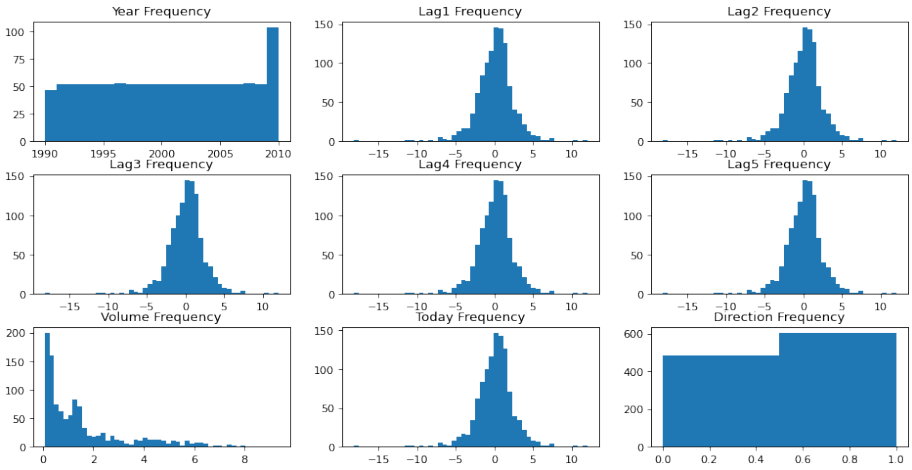
From the covariance matrix we can see that the variance of the "Lag" predictors is roughly the same.
We also see that all Predictors, except for Volume, have a negative relationship with year.

Correlation of the data set:

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
Year	1.000000	-0.032289	-0.033390	-0.030006	-0.031128	-0.030519	0.841942	-0.032460	-0.022200
Lag1	-0.032289	1.000000	-0.074853	0.058636	-0.071274	-0.008183	-0.064951	-0.075032	-0.050004
Lag2	-0.033390	-0.074853	1.000000	-0.075721	0.058382	-0.072499	-0.085513	0.059167	0.072696
Lag3	-0.030006	0.058636	-0.075721	1.000000	-0.075396	0.060657	-0.069288	-0.071244	-0.022913
Lag4	-0.031128	-0.071274	0.058382	-0.075396	1.000000	-0.075675	-0.061075	-0.007826	-0.020549
Lag5	-0.030519	-0.008183	-0.072499	0.060657	-0.075675	1.000000	-0.058517	0.011013	-0.018168
Volume	0.841942	-0.064951	-0.085513	-0.069288	-0.061075	-0.058517	1.000000	-0.033078	-0.017995
Today	-0.032460	-0.075032	0.059167	-0.071244	-0.007826	0.011013	-0.033078	1.000000	0.720025
Direction	-0.022200	-0.050004	0.072696	-0.022913	-0.020549	-0.018168	-0.017995	0.720025	1.000000

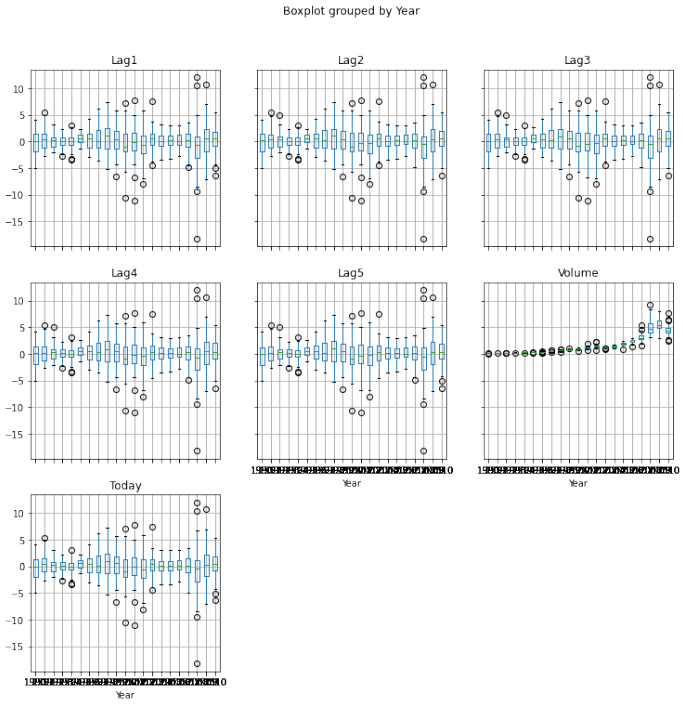
From the correlation matrix we see that most values have a low correlation to

each other, with the only exception being the correlation between "Volume" and "Year" which is high and positive.
Let's look at some histograms of the data:



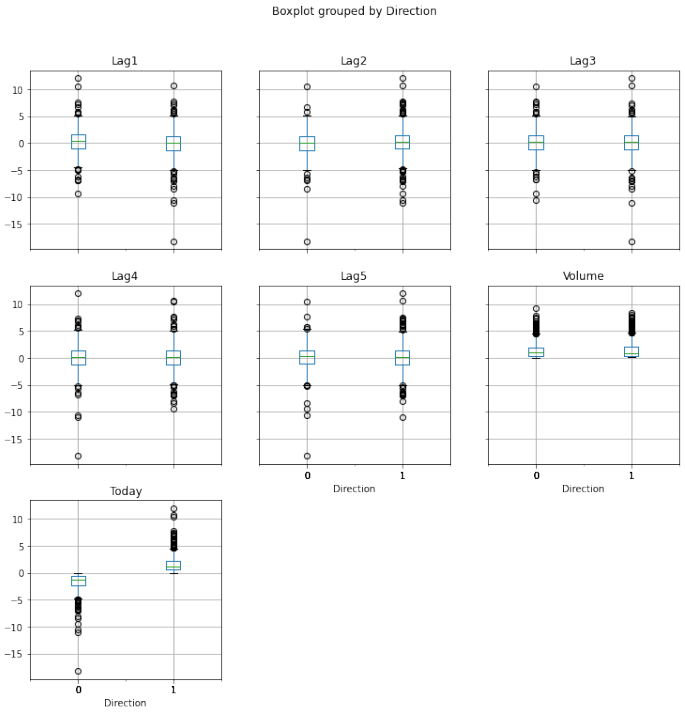
we can see that the "Lag" and Today columns are approximately normally distributed, with center slightly above 0.
The frequency of volume of shares traded decreases exponentially.

Next let's check some box plots with respect to Year:



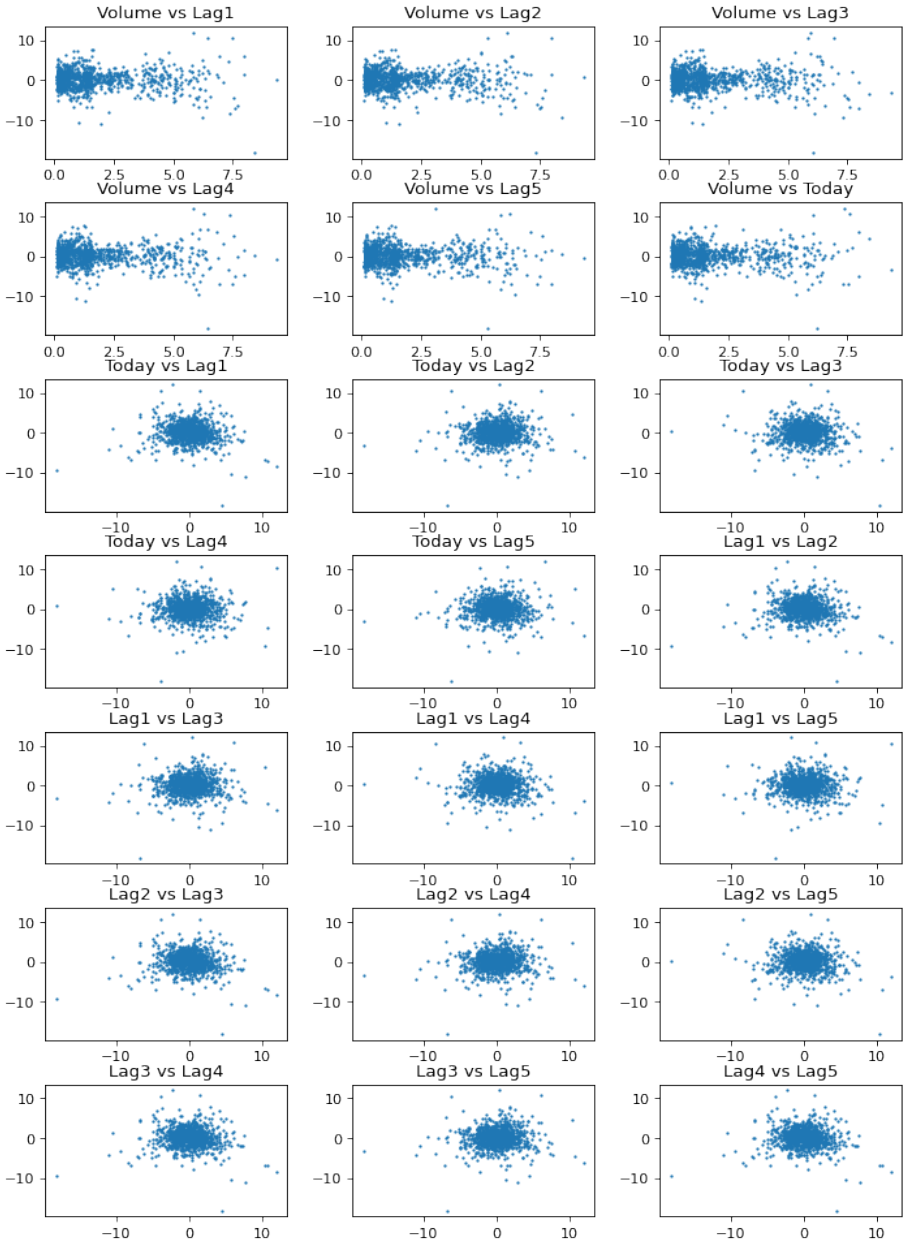
From the box plot we see that the "Lag" indicators and the "today" indicator stay relatively the same each year, with a mean slightly above 0. Their biggest outliers seem to have occurred in the last 5 years.
The volume seems to increase as the years progressed. However, it's variance has also increased.

Let's also check out box plots with respect to Direction:



Here we see that the "Lag" and "Volume" indicators stay relatively the same for either the "Up" or "Down" directions (with "Lag" indicators hovering around 0, and "Volume" hovering around 1).
Expectedly, the "Today" predictors is negative when the direction is "Down" and positive when the direction is "Up".

Finally, let's check out some scatter plots:



Firstly, from scatter plots involving volume, we see that the volume has low correlation with all the other predictors, since it stays relatively constant. Secondly, we can see from scatter plots of "Lag" and "Today" indicators that they also have low correlation with each other.

(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Do any of the predictors appear to be statistically significant? If so, which ones?

Here is a summary of the results:

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	0.266864	0.085930	3.105613	0.001899	0.098445	0.435283
Lag1	-0.041269	0.026410	-1.562610	0.118144	-0.093032	0.010494
Lag2	0.058442	0.026865	2.175384	0.029601	0.005787	0.111096
Lag3	-0.016061	0.026663	-0.602376	0.546924	-0.068320	0.036197
Lag4	-0.027790	0.026463	-1.050141	0.293653	-0.079657	0.024077
Lag5	-0.014472	0.026385	-0.548501	0.583348	-0.066185	0.037241
Volume	-0.022742	0.036898	-0.616333	0.537675	-0.095061	0.049577

Assuming a statistical significance of 0.05, the only predictor for which we reject the null hypothesis is "Lag2" (apart from the constant term). In other words, it is the only predictor which is of any statistical significance.

(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

Here is the confusion matrix:

True	Down	Up
Predicted		
Down	484	605
Up	0	0

	precision	recall	f1-score	support
0	0.444	1.000	0.615	484
1	0.000	0.000	0.000	605
accuracy			0.444	1089
macro avg	0.222	0.500	0.308	1089
weighted avg	0.198	0.444	0.274	1089

The logistic regression never predicts an "Up". Therefore the precision for "Up" is 0. Because of the same reason, the accuracy is equal to the precision for "Down", i.e. 0.444.

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

Here is the confusion matrix:

True	Down	Up
Predicted		
Down	43	61
Up	0	0

	precision	recall	f1-score	support
0	0.413	1.000	0.585	43
1	0.000	0.000	0.000	61
accuracy			0.413	104
macro avg	0.207	0.500	0.293	104
weighted avg	0.171	0.413	0.242	104

Here, again the logistic regression never predicts an "Up". Therefore the overall fraction of correct predictions (i.e. the accuracy) is 0.413.

(e) Repeat (d) using LDA.

Here is the confusion matrix:

True	Down	Up
Predicted		
Down	9	5
Up	34	56

	precision	recall	f1-score	support
0	0.643	0.209	0.316	43
1	0.622	0.918	0.742	61
accuracy			0.625	104
macro avg	0.633	0.564	0.529	104
weighted avg	0.631	0.625	0.566	104

The accuracy is:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{56 + 9}{56 + 9 + 36 + 5} = \frac{65}{104} = 0.625$$

(f) Repeat (d) using QDA.

Here is the confusion matrix:

True	Down	Up
Predicted		
Down	43	61
Up	0	0

	precision	recall	f1-score	support
0	0.413	1.000	0.585	43
1	0.000	0.000	0.000	61
accuracy			0.413	104
macro avg	0.207	0.500	0.293	104
weighted avg	0.171	0.413	0.242	104

The accuracy is:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{0 + 43}{0 + 43 + 0 + 61} = \frac{43}{104} = 0.413$$

(g) Repeat (d) using KNN with K = 1.

Here is the confusion matrix:

True	Down	Up
Predicted		
Down	22	32
Up	21	29

	precision	recall	f1-score	support
0	0.407	0.512	0.454	43
1	0.580	0.475	0.523	61
accuracy			0.490	104
macro avg	0.494	0.494	0.488	104
weighted avg	0.509	0.490	0.494	104

The accuracy is:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{29 + 22}{29 + 22 + 21 + 32} = \frac{41}{104} = 0.394$$

(h) Which of these methods appears to provide the best results on this data?

It seems as though it is the lda, since its accuracy (i.e. proportion of correct predictions) is highest. The other methods seem to fair worse than a random assignment method.

(i) Experiment with different combinations of predictors, including possible

transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

First I have experimented to see whether the interaction between the 'Lag' indicators and the 'Volume' indicator will prove significant for predictions. Therefore the predictors were: Lag1, Lag2, Lag3, Lag4, Lag5, Volume, Lag1*Volume, Lag2*Volume, Lag3*Volume, Lag4*Volume and Lag5*Volume. Here are the results:

LDA

True / Predicted	Down	Up
Down	28	38
Up	15	23

QDA

True / Predicted	Down	Up
Down	31	46
Up	12	15

Logistic Regression

True / Predicted	Down	Up
Down	27	33
Up	16	28

KNN, N=1

True / Predicted	Low	High
Low	19	32
High	24	29

KNN, N=3

True / Predicted	Low	High
Low	15	23
High	28	38

KNN, N=5

True / Predicted	Low	High
Low	16	24
High	27	37

KNN, N=10

True / Predicted	Low	High
Low	21	28
High	22	33

KNN, N=20

True / Predicted	Low	High
Low	23	24
High	20	37

The test error-rate of each method is given below:

$$error - rate_{LDA} = \frac{53}{104} = 0.510$$

$$error - rate_{QDA} = \frac{56}{104} = 0.538$$

$$error - rate_{logit} = \frac{49}{104} = 0.471$$

$$error - rate_{KNN_1} = \frac{56}{104} = 0.538$$

$$error - rate_{KNN_3} = \frac{51}{104} = 0.490$$

$$error - rate_{KNN_5} = \frac{51}{104} = 0.490$$

$$error - rate_{KNN_{10}} = \frac{50}{104} = 0.481$$

$$error - rate_{KNN_{20}} = \frac{44}{104} = 0.423$$

As we can see, these models are sometimes even worse than pure random chance at properly estimating the class of the response variable. The best one was KNN with N=20, which is not saying much, since it's test error rate is 0.423.

Finally I argued that "Lag1" would have the largest influence, and "Lag5" the smallest. Therefore the predictors were: Lag1, Lag2, Lag3, Lag4, Lag5, $Lag1^5$, $Lag2^4$, $Lag3^3$, and $Lag4^2$. Here are the results:

LDA

True / Predicted	Down	Up
Down	30	44
Up	13	17

QDA

True / Predicted	Down	Up
Down	43	54
Up	0	7

Logistic Regression

True / Predicted	Down	Up
Down	6	6
Up	37	55

KNN, N=1

True / Predicted	Low	High
Low	17	30
High	26	31

KNN, N=3

True / Predicted	Low	High
Low	14	24
High	29	37

KNN, N=5

True / Predicted	Low	High
Low	16	24
High	27	37

KNN, N=10

True / Predicted	Low	High
Low	17	30
High	26	31

KNN, N=20

True / Predicted	Low	High
Low	19	23
High	24	38

The test error-rate of each method is given below:

$$error - rate_{LDA} = \frac{57}{104} = 0.548$$

$$error - rate_{QDA} = \frac{54}{104} = 0.519$$

$$error - rate_{logit} = \frac{43}{104} = 0.413$$

$$error - rate_{KNN_1} = \frac{56}{104} = 0.538$$

$$error - rate_{KNN_3} = \frac{53}{104} = 0.510$$
$$error - rate_{KNN_5} = \frac{51}{104} = 0.490$$
$$error - rate_{KNN_{10}} = \frac{56}{104} = 0.538$$
$$error - rate_{KNN_{20}} = \frac{47}{104} = 0.452$$

We get similar results as with the last set of predictors, i.e. most models are worse than random chance. The best algorithm now is logistic regression, with an error rate of 0.413.

In conclusion it seems as though the original set of predictors outlined in part d), along with KNN where N=1, seems to give the best prediction results.

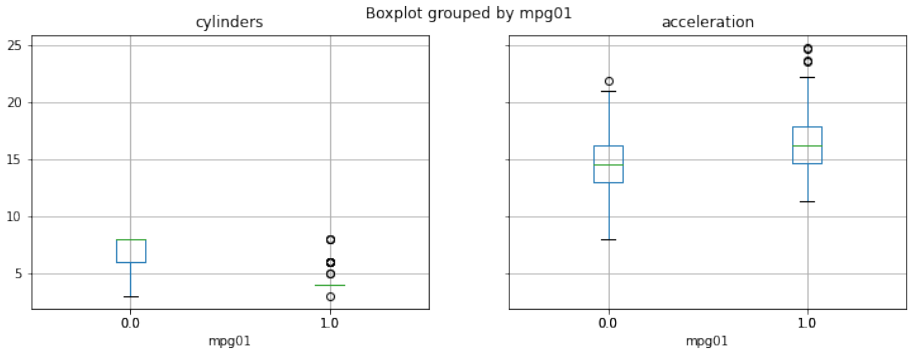
2. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

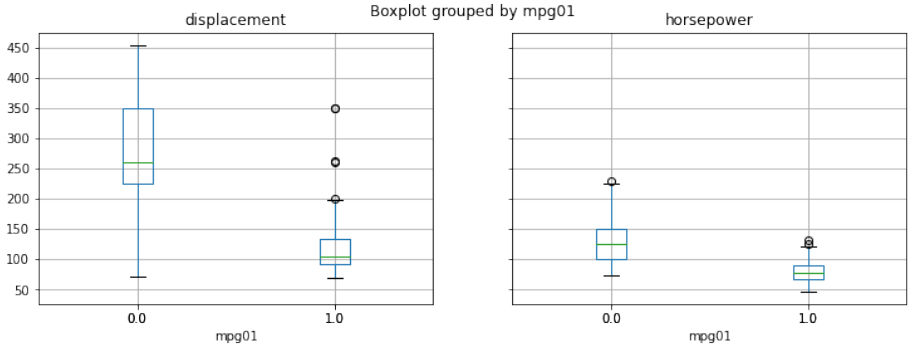
(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. Create a single data set containing both mpg01 and the other Auto variables.

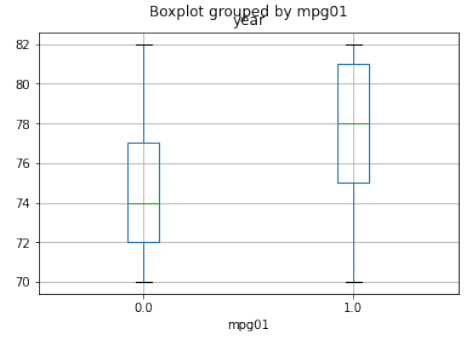
	cylinders	displacement	horsepower	weight	acceleration	year	american	european	japanese	mpg01
0	8	307.0	130	3504	12.0	70	1	0	0	0.0
1	8	350.0	165	3693	11.5	70	1	0	0	0.0
2	8	318.0	150	3436	11.0	70	1	0	0	0.0
3	8	304.0	150	3433	12.0	70	1	0	0	0.0
4	8	302.0	140	3449	10.5	70	1	0	0	0.0
...	...	...	...	...	...	...	...	...	...	...
392	4	140.0	86	2790	15.6	82	1	0	0	1.0
393	4	97.0	52	2130	24.6	82	0	1	0	1.0
394	4	135.0	84	2295	11.6	82	1	0	0	1.0
395	4	120.0	79	2625	18.6	82	1	0	0	1.0
396	4	119.0	82	2720	19.4	82	1	0	0	1.0

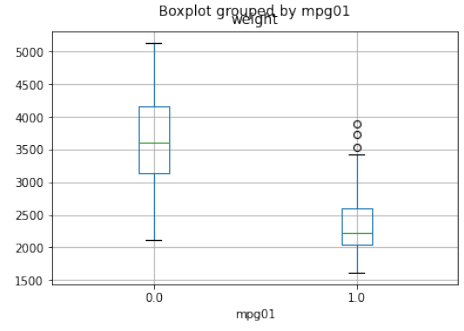
(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

First, let's look at the box plots with respect to mpg01:





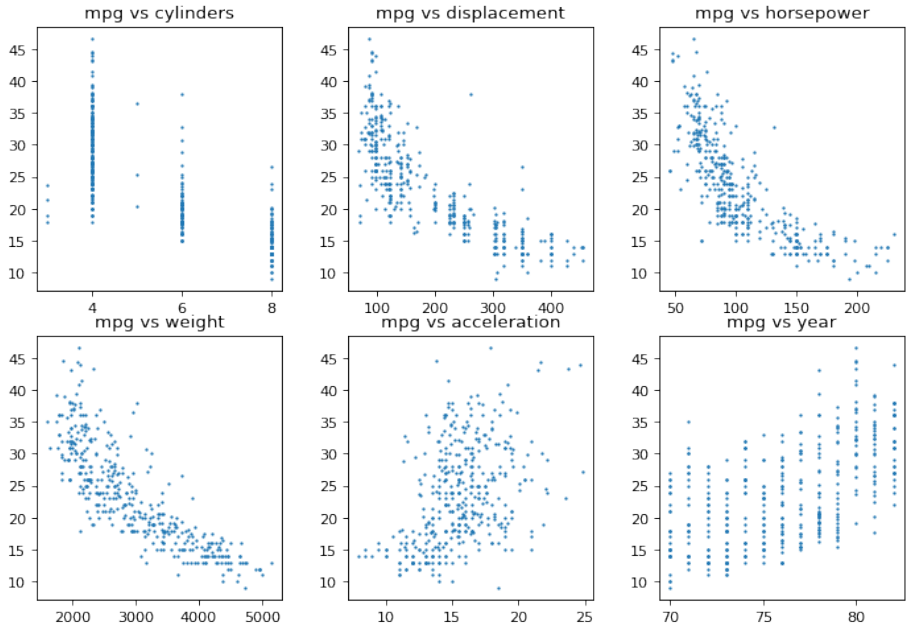




From the box plots we see that most vehicles with high mpg are 4 cylinder, on

average accelerate faster, have a smaller displacement, have lower horsepower, are made in later years and weigh less.

Now let's look at some scatter plots:



Here, we see similar results as seen with box plots: vehicles which are more fuel efficient in general have less cylinders, have smaller displacement, lower horsepower, less weight, accelerate faster and are built more recently.
Based on the above discussion, all 6 predictors will be used to make a prediction about mpg.

(c) Split the data into a training set and a test set.

```
y = data['mpg01']
data.drop('mpg', axis=1)
X = data[['cylinders','displacement','horsepower', 'weight', 'acceleration', 'year']]

X_train, X_test, y_train, y_test = train_test_split(
     X, y, test_size=0.33, random_state=42)

X_train
```

	cylinders	displacement	horsepower	weight	acceleration	year
368	4	112.0	88.0	2640	18.6	82
182	4	107.0	86.0	2464	15.5	76
120	4	121.0	112.0	2868	15.5	73
309	4	98.0	76.0	2144	14.7	80
221	8	305.0	145.0	3880	12.5	77
...	...	...	...	...	...	...
72	8	304.0	150.0	3892	12.5	72
107	6	232.0	100.0	2789	15.0	73
272	4	151.0	85.0	2855	17.6	78
352	4	98.0	65.0	2380	20.7	81
103	8	400.0	150.0	4997	14.0	73

262 rows × 6 columns

(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
True       Low  High
Predicted
Low         54     0
High        16    60
              precision    recall  f1-score   support

         0.0      1.000     0.771     0.871        70
         1.0      0.789     1.000     0.882        60

    accuracy                         0.877       130
   macro avg      0.895     0.886     0.877       130
weighted avg      0.903     0.877     0.876       130
```

The test error rate is:

$$error_rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{16 + 0}{60 + 54 + 16 + 0} = \frac{16}{130} = 0.123$$

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
True       Low  High
Predicted
Low         56     1
High        14    59
              precision    recall  f1-score   support

         0.0      0.982     0.800     0.882        70
         1.0      0.808     0.983     0.887        60

    accuracy                          0.885       130
   macro avg      0.895     0.892     0.885       130
weighted avg      0.902     0.885     0.884       130
```

The test error rate is:

$$error_rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{14 + 1}{59 + 56 + 14 + 1} = \frac{15}{130} = 0.115$$

(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
True       Low  High
Predicted
Low         54     0
High        16    60
              precision    recall  f1-score   support

         0.0      1.000     0.771     0.871        70
         1.0      0.789     1.000     0.882        60

    accuracy                          0.877       130
   macro avg      0.895     0.886     0.877       130
weighted avg      0.903     0.877     0.876       130
```

Performance is same as with LDA. i.e. the test error rate is 0.123

(g) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

KNN wihth 1 neighbor:

True	Low	High
Predicted		
Low	53	3
High	17	57

	precision	recall	f1-score	support
0.0	0.946	0.757	0.841	70
1.0	0.770	0.950	0.851	60
accuracy			0.846	130
macro avg	0.858	0.854	0.846	130
weighted avg	0.865	0.846	0.846	130

Test error rate:

$$error - rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{20}{130} = 0.154$$

KNN with 3 neighbors:

True	Low	High
Predicted		
Low	52	1
High	18	59

	precision	recall	f1-score	support
0.0	0.981	0.743	0.846	70
1.0	0.766	0.983	0.861	60
accuracy			0.854	130
macro avg	0.874	0.863	0.853	130
weighted avg	0.882	0.854	0.853	130

Test error rate:

$$error - rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{19}{130} = 0.146$$

KNN with 5 neighbors:

True	Low	High
Predicted		
Low	49	1
High	21	59

	precision	recall	f1-score	support
0.0	0.980	0.700	0.817	70
1.0	0.738	0.983	0.843	60
accuracy			0.831	130
macro avg	0.859	0.842	0.830	130
weighted avg	0.868	0.831	0.829	130

Test error rate:

$$error - rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{22}{130} = 0.169$$

KNN with 10 neighbors:

True	Low	High
Predicted		
Low	48	1
High	22	59

	precision	recall	f1-score	support
0.0	0.980	0.686	0.807	70
1.0	0.728	0.983	0.837	60
accuracy			0.823	130
macro avg	0.854	0.835	0.822	130
weighted avg	0.864	0.823	0.821	130

Test error rate:

$$error - rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{23}{130} = 0.177$$

KNN with 20 neighbors:

True	Low	High
Predicted		
Low	47	1
High	23	59

	precision	recall	f1-score	support
0.0	0.979	0.671	0.797	70
1.0	0.720	0.983	0.831	60
accuracy			0.815	130
macro avg	0.849	0.827	0.814	130
weighted avg	0.859	0.815	0.812	130

Test error rate:

$$error - rate = \frac{FP + FN}{TP + TN + FP + FN} = \frac{24}{130} = 0.185$$

In conclusion KNN with 3 neighbors seems to perform best on this data set since it has the lowest error-rate.

3. Using the Boston data set, fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA, and KNN models using various subsets of the predictors. Describe your findings.

First, I tried using all of the predictors when constructing the models. Here are the results:

Logistic regression:

True	Low	High
Predicted		
Low	77	25
High	7	58

$$error - rate = \frac{32}{167} = 0.192$$

LDA:

True	Low	High
Predicted		
Low	82	26
High	2	57

$$error - rate = \frac{28}{167} = 0.168$$

QDA:

True	Low	High
Predicted		
Low	84	21
High	0	62

$$error - rate = \frac{21}{167} = 0.126$$

KNN(N=[1,3,5,10,20]):

True	Low	High
Predicted		
Low	79	12
High	5	71
True	Low	High
Predicted		
Low	81	12
High	3	71
True	Low	High
Predicted		
Low	78	12
High	6	71
True	Low	High
Predicted		
Low	75	12
High	9	71
True	Low	High
Predicted		
Low	72	15
High	12	68

$$error - rate_1 = \frac{17}{167} = 0.102$$

$$error - rate_3 = \frac{15}{167} = 0.090$$

$$error - rate_5 = \frac{18}{167} = 0.108$$

$$error - rate_{10} = \frac{21}{167} = 0.126$$

$$error - rate_{20} = \frac{27}{167} = 0.162$$

By comparing the test error rates of the different models, we see that KNN with N=3, outperforms all other models since it has lowest test error rate of 9%. KNN in general seems like the best method to use, followed closely by QDA. Logistic regression seems to fair the worse.

Next, I tried using only the predictors which to me intuitively should be likely predictors of crime rate. Those are: 'nox', 'rm', 'age', 'lstat', 'medv'. Here are the results:

Logistic regression:

True	Low	High
Predicted		
Low	69	14
High	15	69

$$error - rate = \frac{29}{167} = 0.174$$

LDA:

True	Low	High
Predicted		
Low	77	24
High	7	59

$$error - rate = \frac{31}{167} = 0.186$$

QDA:

True	Low	High
Predicted		
Low	76	26
High	8	57

$$error - rate = \frac{32}{167} = 0.192$$

KNN(N=[1,3,5,10,20]):

True	Low	High
Predicted		
Low	66	21
High	18	62
True	Low	High
Predicted		
Low	70	19
High	14	64
True	Low	High
Predicted		
Low	72	19
High	12	64
True	Low	High
Predicted		
Low	72	20
High	12	63
True	Low	High
Predicted		
Low	71	19
High	13	64

$$error - rate_1 = \frac{39}{167} = 0.234$$

$$error - rate_3 = \frac{33}{167} = 0.198$$

$$error - rate_5 = \frac{31}{167} = 0.186$$

$$error - rate_{10} = \frac{32}{167} = 0.192$$

$$error - rate_{20} = \frac{32}{167} = 0.192$$

As the number of predictors decreased, the accuracy of all models decreased. All models seem to perform at about the same test error rate between 18.6% and 19.8%. The exception is logistic regression, which now fairs best at 17.4% test error rate, and KNN with N=1, with the highest test error rate of 23.4%.

Finally, I tried using only the predictors which I've excluded in the previous step. Those are: 'zn','indus', 'dis', 'rad', 'tax', 'ptratio' and 'black'. Here are the results:

Logistic regression:

True	Low	High
Predicted		
Low	75	23
High	9	60

$$error - rate = \frac{19}{167} = 0.174$$

LDA:

True	Low	High
Predicted		
Low	79	27
High	5	56

$$error - rate = \frac{32}{167} = 0.192$$

QDA:

True	Low	High
Predicted		
Low	84	20
High	0	63

$$error - rate = \frac{20}{167} = 0.120$$

KNN(N=[1,3,5,10,20]):

True	Low	High
Predicted		
Low	73	5
High	11	78

True	Low	High
Predicted		
Low	75	9
High	9	74

True	Low	High
Predicted		
Low	75	8
High	9	75

True	Low	High
Predicted		
Low	70	8
High	14	75

True	Low	High
Predicted		
Low	67	11
High	17	72

$$error - rate_1 = \frac{16}{167} = 0.096$$

$$error - rate_3 = \frac{18}{167} = 0.108$$

$$error - rate_5 = \frac{17}{167} = 0.102$$

$$error - rate_{10} = \frac{22}{167} = 0.132$$

$$error - rate_{20} = \frac{28}{167} = 0.168$$

By comparing the test error rates of the different models, we see that KNN with N=1, outperforms all other models since it has lowest test error rate of 9,6%. KNN in general seems like the best method to use, followed closely by QDA. LDA seems to fair the worse. To my surprise the predictors which I thought would not be good for predicting turned out to be better than the ones I thought would be.

By comparing all models by their test error rate, we see that the best one is KNN with N=3 which includes all predictors, which only narrowly beats the second best method of KNN with N=1 from the last set of predictors. The interesting difference between them is that the best method has a higher false negative rate than false positive rate ($\frac{12}{93} = 0.129$ compared to $\frac{3}{74} = 0.041$) than the second best method whose false negative rate is lower than its false positive rate ($\frac{5}{78} = 0.064$ compared to $\frac{11}{89} = 0.124$)