Machine Learning & Predictive Analytics
UFCFMJ-15-M
20042768

SECTION A

1. [10 points] Linear Regression models can be optimised to avoid high variance by means of regularizations such as Lasso and Ridge. Explain when and why they are applied and the difference between both.

When finding the optimal training line for a Linear Regression model, the presence of variance in the test set is inevitable. In the absolute simplest way possible, variance refers to how spread out the data may be given the mean of the set. Because of this issue, the regularization parameters of Lasso and Ridge are tools which apply penalties to our residual points. The mathematical formulas are very similar to each other, but differ on one simple fundamental. Ridge regression penalizes the sum of squared coefficients, and Lasso penalizes the sum of their absolute values.

Inherently, because of this fundamental difference Lasso narrows down the features with its formula, and for that reason it performs better when you have only a handful of features which have greater impact on the model. Lasso does a good job of eliminating or minimizing certain features which already had a significantly small variance to begin with. Hence the name Least Absolute Shrinkage and Selection Operator (LASSO), because the formula provides a narrow focus on the more impactful varying data, and penalizes them appropriately, it is a good method to choose when there are less significant features.

Ridge regression keeps all features intact, and applies the penalization parameter in a manner where all features are still represented. Therefore Ridge is the best choice when there is a dataset with many parameters that are very similar to each other. They have a high multicollinearity value, and their impact in predicting one another is fairly consistent.

2. In the context of learning a predictive model, explain the following:
    a. [7 points] What does it mean to ‘fit’ a model.
    b. [7 points] Explain the Bias-Variance tradeoff.
    c. [7 points] Discuss how cross validation is used to evaluate model performance.
    d. [9 points] Explain what the Coefficient of Determination (R2; R squared) is, and explain how the measure is used for Linear Regression models.

a. When one ‘fits’ a model on a dataset, they are providing the data to a given algorithm. Once that data is digested, and the model has the information there are many different parameters, and hyperparameters to deploy the best model possible. One thing that should be mentioned is people typically do not want to fit their model too closely to

the dataset. Reason being is that this is a recipe for overfitting. If your model learns the provided data very specifically, then it will not be able to generalize and predict other data very well. This example was provided in class when we discussed the ability to classify an orange by dropping it into a mold. There are many different parameters that play a part in this instance. From how far is the orange being dropped, how much does the orange way, is it being thrown with force? All of these questions could be variables in our dataset, and the more information we have that aligns with consistent outcomes gives our model a greater ability to generalize unseen information. Therefore, when we fit a model we have many different tools to consider with each different algorithm. At the end of the day, we want to fit our given model with the optimal parameters, and hyperparameters that best predict our target variable.

b. The combination of bias and variance is a well-known tandem in the machine learning world, and is what prevents models from absolute dominance. To provide an example, a model with high variance, and low bias would be considered as overfitting. Therefore when trying to predict our target variable on an unseen test set, the accuracy would be very poor, but high on the training set. This is commonly expressed on a dart board, and could be represented by points scattered at various points relatively around the bullseye.

Vice versa if we had high bias, and low variance when fitting the model this would result in underfitting. Reason being is that the algorithm fails to understand the underlying relationships between features and target variables. As an illustration on a dart board this would be hitting a consistent area on the board, but not the area that we are aiming for. The machine learning model does not understand the aiming point, and results in high error on training and test set.

The tradeoff in finding the optimal value between bias and variance is a tricky one. The first step is to find out where the issues are occurring. If the model is having high bias some of the solutions to solve are as follows: Add more variables to help the algorithm understand the data better, increase complexity, or decrease regularization. For high variance the opposite can be said; reduce input variables and only use relevant variables, or simply get more training data. The ideal model reduces bias and variance while also maintaining a decent amount of complexity. This is the basis behind machine learning in producing expected outcomes on unseen data, while also comprehending the training data.

c. Cross validation is an extremely useful tool to make full use of a data set, and provide multiple outcomes for various data segments. Cross validation can be used to evaluate model performance by breaking the given data into explicit folds. For example, if the amount of folds chosen was five for a dataset of 10000, then the dataset would be broken down into five equal folds of 2000.

The way the method works is that it is evaluated on a singular different fold each time. Therefore the data is fit on 4 folds (8000 records), and then tested on the fifth. The process will continue until each fold has been tested upon. This allows us to be more confident in our results, as it is not overfitting on a certain portion of our training data,

and we can get a better general outlook. Another benefit of cross validation in evaluating a model is that you can create a nested for loop to find optimal parameters. Instead of a repetitive guess and check this provides results all in one go, with a method of directly returning best parameters.

d. The coefficient of determination or R squared is a vital measure which takes advantage of the correlation between two variables on a linear scale and compares them to the mean in order to describe the variation within a relationship. In linear regression models we get multiple variables which provide an accurate measure of predicting one another. When we plot on a linear scale, and apply the R squared metric we can truly see if two variables are responsible for the variation compared to our baseline mean value.

   For example, if we were attempting to describe a linear relationship that assesses how much time is spent studying with regards to how much sleep is gotten. Say the R squared metric comes out to 95%, then this means that this is a huge indicator for correlation. This means that the fitted line would have the residuals fitting almost perfectly to it, as the sleep/study relationship accounts for 95% of the variation.

SECTION B

5. You are given a dataset below to learn a decision tree which predicts if students pass the machine learning course (Yes or No), based on previous students performance in the Programming for Data Science course (High 'H', Medium 'M', or Low 'L') whether or not they studied (True or False) and whether they studied early in the day (E), late (L) or Night (N):

Grade in Programming for DS	Studied	Study hour	Passed (Target)
H	T	E	T
H	F	E	T
L	T	E	T
L	F	E	F
M	T	E	T
M	F	E	F

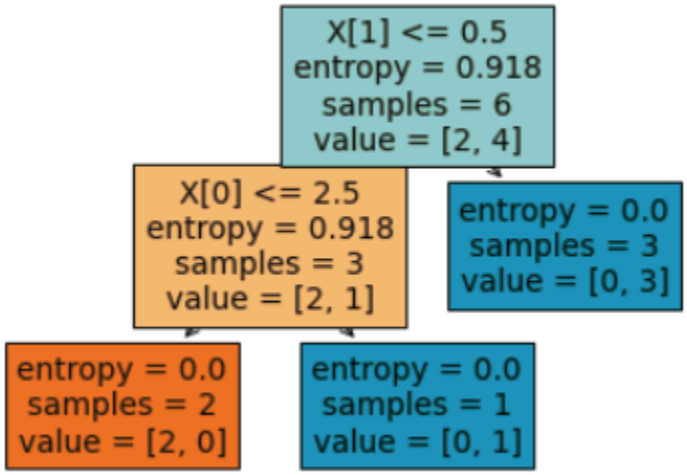
As in the ID3 and C4.5 algorithms where the Information Gain and Entropy methods are used to infer Decision Trees, you are asked to solve the following:

a. [6 points] What is the Entropy H (Passed) and H (Passed | Grade in Programming for DS). You can write your answers using log base 2, but it may be helpful to note that log base 2 3 ~ 1.6.

= - (4/6 log 2 (4/6) + 2/6 log 2 (2/6))
The entropy for Passed as a whole was .90

The entropy for passed when partitioning on Grade in Programming for DS was .918 as shown below.

b. [10 points] Draw the 'optimal' decision tree (optimal on training) fully and justify your choice with a features importance ranking.



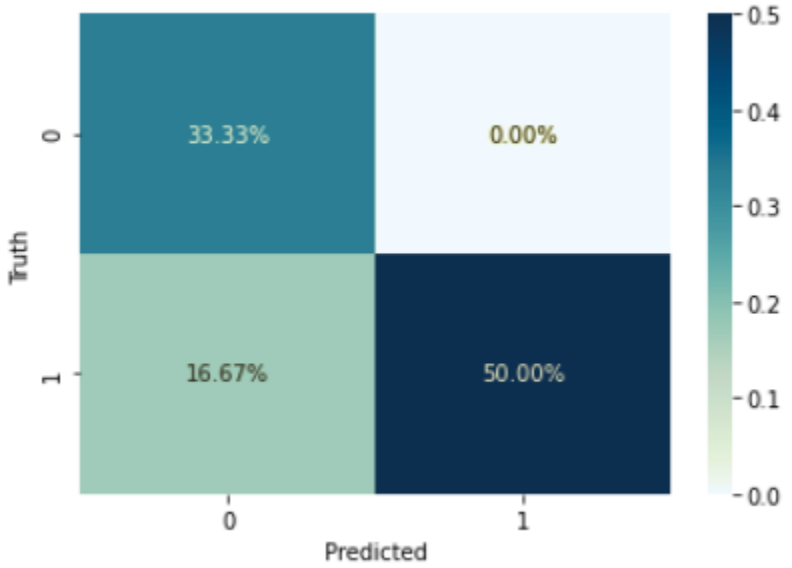
We initially split on X[1] also known as the studying variable for our root node. So in other words, the studying variable provided us with the best possible entropy. The studying variable gave us the best possible purity results when we classified our pass/fail split on that initial variable. However, the Programming for Data Science variable came back with an equal entropy score giving us the same information gain.

From that point we had two possible variables left, which were Grade For Programming in Data Science and Study Hour. Study hour proved to be irrelevant, as it was not considered to be split on, and all outcomes had the same study hour. Therefore out of the two remaining variables, Grade for Programming in Data Science was the next best classifier for entropy levels that were below 0.5. Also I want to point out that for entropy levels greater than 0.5 at the root node the split was solidified as a leaf node. The split had 100% purity as it had all of the same class labels. In the end, all split had 100% pure class labels resulting in 3 leaf nodes.

Disclaimer - Letter columns had to be transformed into numbers.

c.

- **1 for Pass, and 0 for fail**



- The model correctly predicted 33.33% of students who failed.
- The model correctly predicted 50% of students who passed.
- The model incorreclty predicted 16.67% students who passed as predicted fails.

	precision	recall	f1-score	support
0	0.67	1.00	0.80	2
1	1.00	0.75	0.86	4
accuracy			0.83	6
macro avg	0.83	0.88	0.83	6
weighted avg	0.89	0.83	0.84	6

- The accuracy of this decision tree is 83%, and the f1-scores are shown for each specific class.
- Considering how much data was present for this algorithm, the scores are sufficient.

- General limitation of a decision tree:

1. Not built for large datasets (Random Forest instead(
2. Decision trees can overfit to the training data, which can lead to poor accuracy on test sets, yet good accuracy on training.
3. Algorithms are more complex compared to others, and can potentially become too complex.

D. [4 points] Discuss how Decision Trees performance can be improved by a method of ensemble learning.

Random Forest presents the opportunity to create an algorithm from multiple decision trees. These can span from a few to hundreds. The power of ensemble learning harnesses the power of groupthink, and submits final decisions as an aggregate. The benefits of ensemble methods typically result in better, more accurate predictions, while also narrowing down the variance of a model. One may ask how ensemble methods utilize a group think approach. Well the way that they do it is by creating multiple decision trees to learn alternative mapping functions for solving the same issue. The trees assess different variables, and their likelihood of predicting outcomes are very different. Through this way the errors that are present in each decision tree are taken into account, and by considering all scenarios better predictions are output. More specifically ensembles for regression assess hyperplanes, while classification relies on decision boundaries.

4. Naïve Bayes Classifier is one of the simplest learning algorithms that is widely used in the context of text classification. Explore the fundamental concepts and design of the method in an application to classify SPAM e-mails from Non-SPAM, or an application of Sentiment Analysis (with *positive* and *negative* being the possible classes). In your exploration make sure to consider the following:
    a. [10 points]. Explain how the method works.
    b. [5 points]. Main assumptions of the Bayesian method for classification.
    c. [5 points]. Main limitations of the method.
    d. [10 points]. Is there a loss function for a Bayes-inferred classifier? Explain if your answer is yes or justify it with example otherwise (eg., in comparison with another classifier's loss function or approach you are familiar with).

a. In an application of a spam classifier utilizing Naive Bayes it begins with narrowing down words (training set). Between spam, and non spam emails we look to find common ground between the two and come up with words that are relevant. Before we get into our selected words we calculate the Prior Probability of each class given our training set, or we can estimate this value. From this calculation we see the general probability of Spam and Non Spam occurring given our dataset.

   Once we have this number we can begin to calculate the individual numeric occurrence of each specified word relative to each class. Once we see the total instances of each individual word we multiply everything together with the general probability.

   The formula is the probability of our class * probability of each of our given words | class.

In this way our calculation is compared to the output of Spam and Non Spam with the greater value receiving the target label.

b. The main assumption of the Bayesian method for classification is that the algorithm assumes that the words do not have syntax, and features are unrelated. When in reality, the way that we structure sentences as humans are very impactful, and play a big role in what word will be coming in next. Naive Bayes does not take this into account, and assumes that all features are independent and unrelated to one another. The computer has no way of understanding context, grammar, emphasis, etc. There is also no way of learning from previous results in creating a spam protector.

c. One of the main limitations in this method is known as “Zero Frequency”. This is an issue where we train this model on test words, and if that word did not occur in our given class then it will completely misconstrue the outcome. Therefore we must use a hyperparameter known as alpha which makes sure that we account for all words in our set for at least one occurrence.

   Another limitation that does not apply to the Spam protector would be overgeneralizing continuous values through histograms. It could be easy to lose data when selecting the general frequency of continuous bins.

d. There is no loss function for a Naive Bayes classifier. Naive Bayes operates off of frequency, and the posterior probability plays a huge role in doing so. The algorithm makes probable decisions based on the training set, and is more likely to classify based off of prior frequencies. Therefore the emphasis is on these weights, and has no other prior knowledge otherwise. When we compare this to a different loss function such as absolute error loss we have information for the predicted and the actual values. Therefore we are able to measure what was actually lost. If Naive Bayes had a gauge to measure what it was losing it would no longer be Naive.