Exam PA December 13, 2019 Project Report Template

Instructions to Candidates: Please remember to avoid using your own name within this document or when naming your file. There is no limit on page count.

As indicated in the instructions, work on each task should be presented in the designated section for that task.

> *This model solution is provided so that candidates may better prepare for future sittings of Exam PA. It includes both a sample solution, in plain text, and commentary from those grading the exam, in italics. In many cases there is a range of fully satisfactory approaches. This solution presents one such approach, with commentary on some alternatives, but there are valid alternatives not discussed here.*

Task 1 – Examine each variable and make appropriate adjustments (12 points)

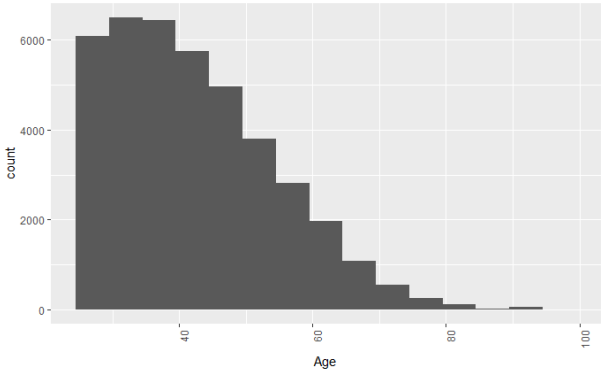
Candidates were expected to analyze and comment on each variable's univariate distribution and its bivariate distribution with the target variable. Commenting on a selection of the variables only received partial credit. Most candidates correctly excluded applicants under the age of 25. Many candidates only included minimal commentary about the variables. Better candidates commented on data anomalies and the frequency of high/low values in the target variable. Data adjustments needed to be accompanied by valid reasons.

The dataset contains 48,842 observations on eight variables. One, value_flag, is the target variable. It is a zero-one variable with 11,687 (24%) equal to one, which represents high value. After removing the observations with age less than 25, there are 40,410 observations, with 29% high value.

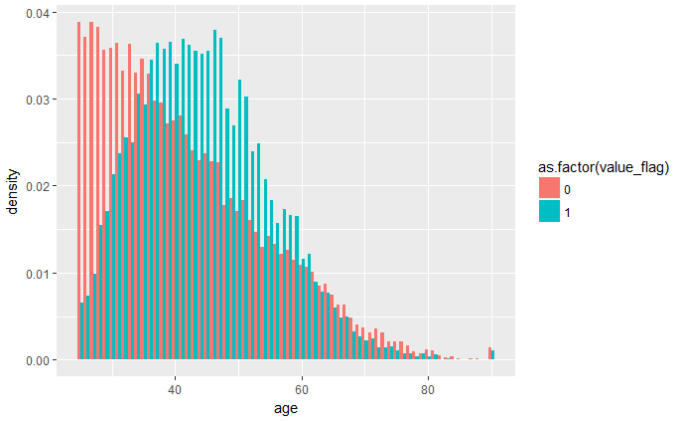
I will explore six of the seven potential predictor variables. The variable cap_gain will be examined later.

age

Ages are integers from 25 to 90. The distribution is slightly right-skewed, but otherwise shows nothing unusual such as outliers or abnormally high/low density regions.

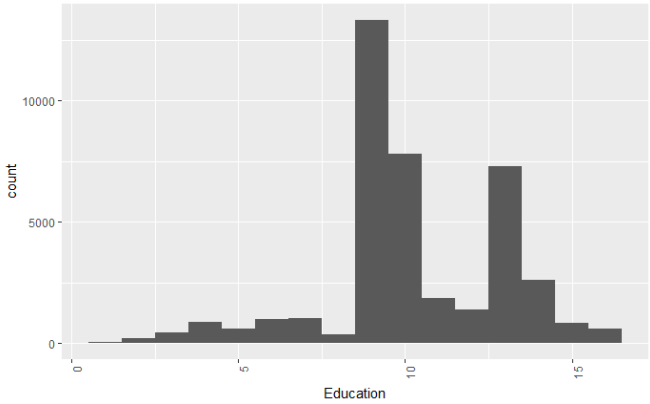


The following graph shows that older people are more likely to be high value, except perhaps at older ages. This should be a good predictor.

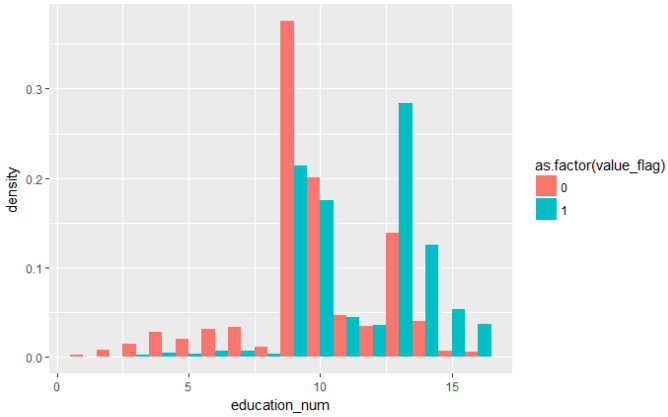


education_num

This is an integer between 1 and 16. All observations are in this range. The following histogram shows that there are a few surprisingly low values, but I have no reason to question or eliminate them. Because the relationship of these numbers to actual education accomplishments is unknown, it is not clear what the values with high frequency mean.

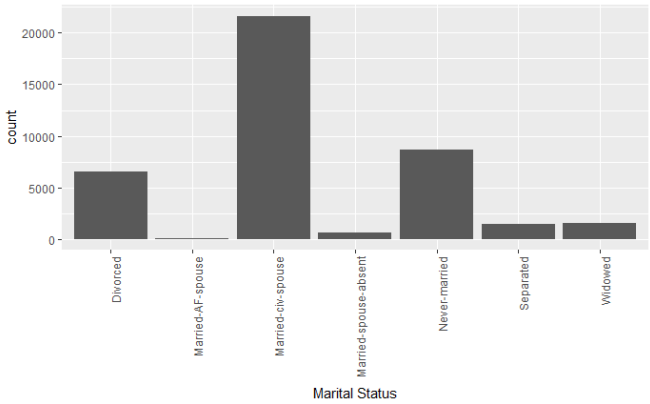


The following graph shows that those with more education are more likely to be of high value.



marital_status

There are seven levels for this variable. The following histogram shows that several have small counts.



Looking at the proportion of ones for value_flag (below), I see that martial status makes a large difference. I note that the two married with spouse present levels have fairly similar (large) proportions. With only 31 observations in one of them, those two will be combined. I will also set married with spouse as the base level as it is most common.

> *Many candidates reduced the levels of the marital_status variable without providing reasonable explanations for their decisions. Better candidates considered the proportion of high value when reducing the number of levels.*

marital_status	target = low value	target = high value	n	proportion high value
Divorced	5829	669	6498	0.10295476
Married-AF-spouse	19	12	31	0.38709677
Married-civ-spouse	11727	9934	21661	0.45861225
Married-spouse-absent	515	58	573	0.10122164
Never-married	8000	697	8697	0.08014258
Separated	1342	96	1438	0.06675939
Widowed	1384	128	1512	0.08465608

occupation

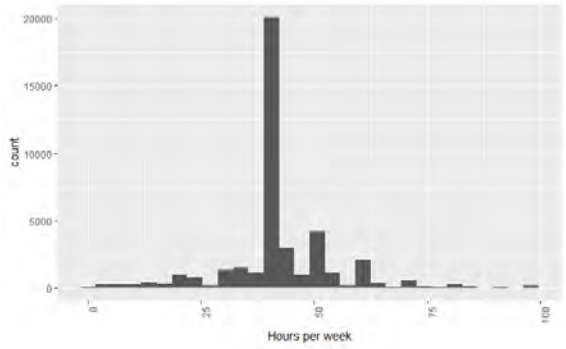
Occupation has been combined into six groups, one of which is a group where the occupation is not available. There are 1,775 in this group and it may be best to leave them as their own level. Because our goal is to make predictions, we need to be able to do so for those whose occupation is unknown. All the levels have reasonable counts.

It is clear from the table below that increasing occupation group numbers relate to an increasing proportion of ones. Due to that relationship it makes sense to keep Group 1 as the base.

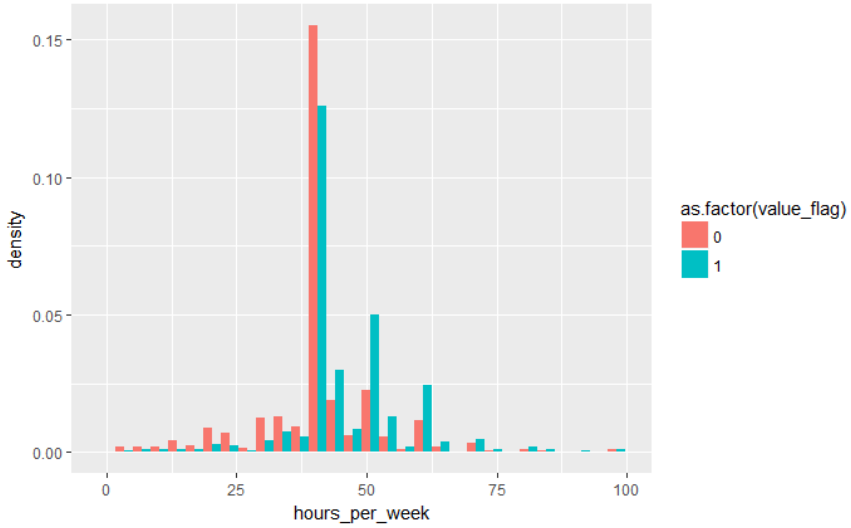
occupation	target = low value	target = high value	n	proportion high value
Group 1	4594	333	4927	0.06758677
Group 2	6951	1300	8251	0.15755666
Group 3	8545	3311	11856	0.27926788
Group 4	1420	725	2145	0.33799534
Group 5	5793	5663	11456	0.49432612
Group NA	1513	262	1775	0.14760563

hours_per_week

These are whole numbers from 1 to 99. The histogram below shows there are a few unusually large and small values, but the data is fairly reasonable when you consider that some individuals might work multiple jobs and some might only have part-time work. It is notable that there are no observations with 0 hours worked, which I would expect to see for unemployed or retired individuals. I will use the data as is, but I recommend contacting the source of the data for further investigation.

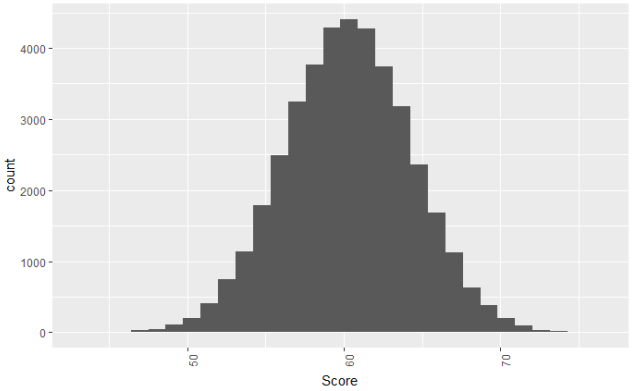


The graph below indicates that more hours per week leads to higher value.

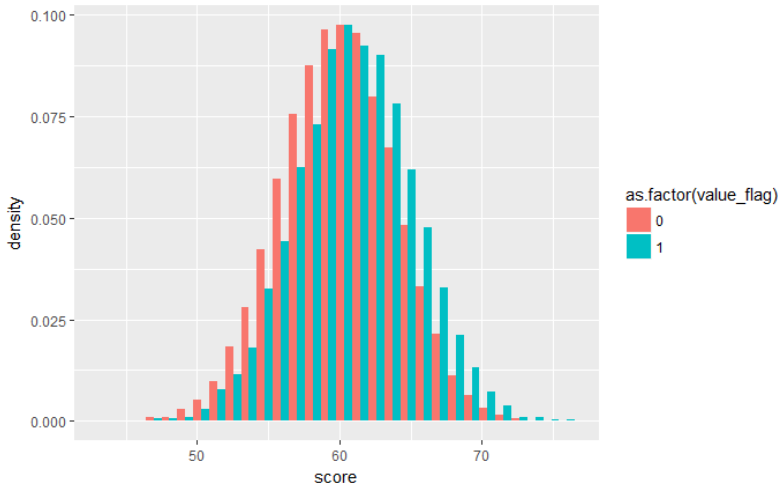


score

This is a number with two decimal places running from 43.94 to 76.53. The histogram below suggests the score is normally distributed and no adjustments are required.



The graph indicates a large overlap for low and high value policyholders, but with higher scores more likely to be high value, but not by much.



In summary, all the variables appear to have predictive power and only slight modifications were required.

Task 2 – Construct a classification tree (10 points)

Most candidates successfully produced a tree and included it in their report, but many failed to adequately compare the trial-and-error approach to the cross-validation approach for setting the cp parameter. In order to earn full credit, candidates needed to build at least two trees using trial and error and justify their parameter decisions. Some candidates interpreted the variables, rather than the tree itself.

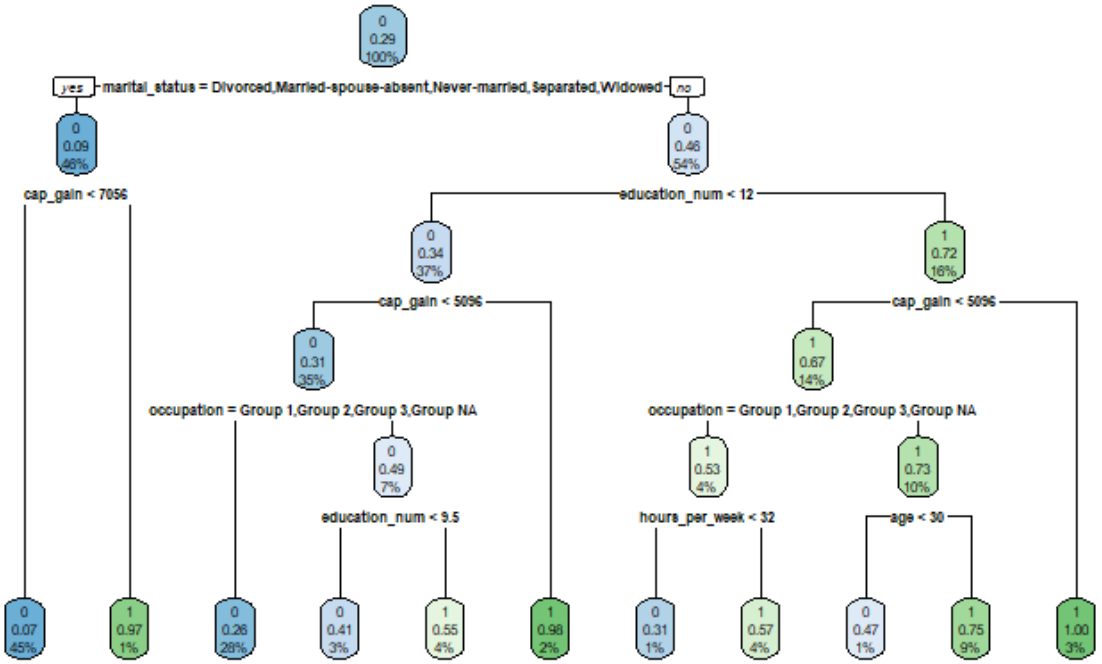
Before constructing a tree, I split the data into training (70%) and testing (30%) sets. The proportion of ones in the two sets were 28.6% and 28.8% respectively, so the stratification appears to have been successful.

I was permitted to alter the minbucket, cp, and maxdepth parameters. Each of these parameters can be used to prevent overfitting, so I will check for that by comparing the training set performance to the test set performance after each adjustment. The "minbucket" parameter sets the smallest size for any leaf node in the tree. The "cp" parameter sets the minimum improvement needed in order to make a split. The "maxdepth" parameter sets the minimum number of levels for the splits. The table below displays the results of each model iteration.

Approach	Model Name	maxdepth	cp	minbucket	Terminal Nodes	Train AUC	Test AUC
Trial & Error	Model 1	6	0.001	5	15	0.8391	0.8448
	Model 2	5	0.001	5	11	0.8382	0.8443
	Model 3	4	0.001	5	5	0.8197	0.8283
	Model 4	5	0.002	5	10	0.8375	0.8442
	Model 5	5	0.005	5	7	0.8321	0.8393
Cross Validation	Model 6	6	0.000	5	22	0.8566	0.8604
	Model 7	5	0.000	5	13	0.8384	0.8444

I initially adjusted the parameters using manual trial and error. I chose to adjust the maxdepth parameter, followed by cp, and finally minbucket. With maxdepth = 6, cp = 0.001, and minbucket = 5. The area under the ROC curve (AUC) is 0.8391 for the training set and 0.8448 for the testing set. This indicates there is no overfitting. However, I think a simpler model should be checked out. Changing maxdepth to 5 only lowers the testing AUC to 0.8443 and the tree is easier to interpret. At maxdepth = 4 AUC drops to 0.8283, which does not seem worth it. I will use maxdepth = 5.

Raising cp to 0.002 does not result in a much simpler tree, and raising cp to 0.005 caused the AUC to decrease too much. I will stick with maxdepth = 5 and cp = 0.001. The terminal nodes of the tree are all larger than the current minbucket setting, and there is no evidence of overfitting. Hence, I chose not to adjust the minbucket parameter. The manually tuned tree selected is Model 2:



An alternative is to estimate the cp parameter using cross-validation. This approach involves dividing the training data up into folds, training the model on all but one of the folds, and measuring the performance on the remaining fold. This process is repeated to develop a distribution of performance values for a given cp value. The cp value that yields the best accuracy is then selected. Even though I'm training the cp parameter with cross validation, the maxdepth parameter is still in play.

Model 6 started with maxdepth equaling 6, and the cp value selected by cross-validation was 0, resulting in the most complex tree yet. The AUC is 0.8604.

Changing maxdepth to 5 for Model 7 yielded an AUC of 0.8444, and a selected cp value of 0. This model is slightly more complex than Model 2 (13 terminal nodes versus 11). For the same reasons as in the manual tuning process, there is no reason to adjust the minbucket parameter. Given the similar AUC values I will choose Model 2, which has maxdepth = 5, cp = 0.001, and minbucket = 5.

The selected tree uses the following variables for making splits:

- marital_status
- education_num
- cap_gain
- occupation
- hours_per_week
- age

The only variable not used is score.

The actual splits are:

```
1) root 28287 8103 0 (0.713543324 0.286456676)
   2) marital_status=Divorced,Married-spouse-absent,Never-
married,Separated,Widowed 13098 1184 0 (0.909604520 0.090395480)
     4) cap_gain< 7055.5 12785  880 0 (0.931169339 0.068830661) *
     5) cap_gain>=7055.5 313    9 1 (0.028753994 0.971246006) *
   3) marital_status=Married-spouse 15189 6919 0 (0.544472974 0.455527026)
     6) education_num< 12.5 10560 3587 0 (0.660321970 0.339678030)
      12) cap_gain< 5095.5 10039 3074 0 (0.693794203 0.306205797)
        24) occupation=Group 1,Group 2,Group 3,Group NA 7936 2042 0
(0.742691532 0.257308468) *
        25) occupation=Group 4,Group 5 2103 1032 0 (0.509272468 0.490727532)
          50) education_num< 9.5 887  360 0 (0.594137542 0.405862458) *
          51) education_num>=9.5 1216  544 1 (0.447368421 0.552631579) *
      13) cap_gain>=5095.5 521    8 1 (0.015355086 0.984644914) *
     7) education_num>=12.5 4629 1297 1 (0.280190106 0.719809894)
      14) cap_gain< 5095.5 3901 1295 1 (0.331966163 0.668033837)
        28) occupation=Group 1,Group 2,Group 3,Group NA 1211  569 1
(0.469859620 0.530140380)
          56) hours_per_week< 32.5 174   54 0 (0.689655172 0.310344828) *
          57) hours_per_week>=32.5 1037  449 1 (0.432979749 0.567020251) *
        29) occupation=Group 4,Group 5 2690  726 1 (0.269888476 0.730111524)
          58) age< 29.5 193   91 0 (0.528497409 0.471502591) *
          59) age>=29.5 2497  624 1 (0.249899880 0.750100120) *
      15) cap_gain>=5095.5 728    2 1 (0.002747253 0.997252747) *
```

Looking at the six nodes that lead to a prediction of 1, we see that the high value customers are those which

1. Are not married with spouse present and capital gains 7056 or more
2. Are married with spouse present, have education less than 10, have capital gains less than 5096, and are in occupation groups 4 or 5
3. Are married with spouse present, have education less than 13 and have capital gains 5096 or more
4. Are married with spouse, have education 13 or more, have capital gains less than 5096, are not in occupation groups 4 or 5, and work at least 33 hours per week
5. Are married with spouse, have education 13 or more, have capital gains less than 5096, are in occupation groups 4 or 5, and are age 30 or more
6. Are married with spouse, education 13 or more, and have capital gains at least 5096.

This can be a bit confusing, but the evidence is that in general, high value comes from being married with spouse present (unless capital gains is low), having education of 13 or more, and being in occupation groups 4 or 5.

Finally, I'll note that there may be an interaction between marital status and education. This is evident because of the nested nature of the splits associated with those variables in the tree. For those that are married with a spouse present, education makes a difference, for others it does not.

Task 3 – Construct a boosted tree (6 points)

Most candidates were able to produce a variable importance plot and include it in their report, but many failed to demonstrate an understanding of variable importance in the context of a boosted tree.

Fitting the boosted tree to the training set and predicting with the test set gave an AUC of 0.8991, which is promising.

I evaluated the importance of each predictor variable and got the following results:

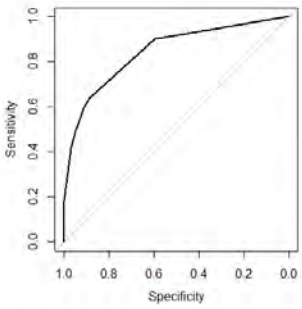
	Importance
cap_gain	100.000000
education_num	72.334565
marital_statusNever-married	47.334402
marital_statusDivorced	37.605056
age	37.055246
score	33.761356
hours_per_week	30.977818
occupationGroup 5	20.704170
marital_statusSeparated	10.278117
marital_statusWidowed	7.506715
occupationGroup 3	4.659374
marital_statusMarried-spouse-absent	3.875703
occupationGroup 4	3.432978
occupationGroup 2	2.203016
occupationGroup NA	0.000000

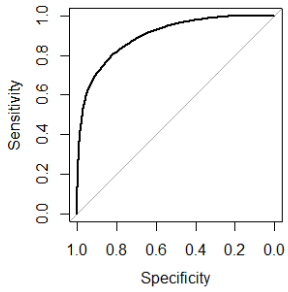
The results are scaled to 100, so the numbers indicate relative importance. The algorithm used evaluates each level of a factor variable separately. Generally, variables that are used to make splits more frequently and earlier in the trees in the boosted trees are determined to be more important. Clearly cap_gain and education_num are much more important than the other variables. In total, the marital_status and occupation variables also contribute to the predictions. Unlike the tree constructed earlier, score is viewed as important. This may be due to the fact that boosting allows variables that would normally be overshadowed by other variables to be fit to the errors made using the other variables.

Task 4 – Recommend a tree model (4 points)

Candidates needed to clearly recommend a tree model, and most did that successfully. Most candidates failed to interpret the ROC curve.

There are two candidate tree models. A single tree was built and had an AUC against the test set of 0.8443. A boosted tree had an AUC of 0.8991. The ROC curves are below, with the single tree first.





The single tree has the advantage of being easy to interpret. When making predictions for future applicants, a series of if/then statements will lead directly to a recommendation. It is also clear how the predictor variables relate to the outcome. The boosted tree is opaque, the predictor variables are put in and a prediction comes out.

Because our goal is to make good predictions, interpretation is less important. As a result, I recommend choosing the model with the better predictive ability, the boosted tree.

Because decisions were made using ROC and AUC, a word of explanation is in order. A classification tree does not return an exact prediction of what group a new individual belongs to, but rather a probability. Specifically in our trees, the predictions are the probability of being in the high value category. For example, in the single tree, if you end up at the third node from the right in the bottom row, the result is a probability of 47% of being high value. If the cutoff for being a high value is 50%, then these observations would be predicted as low value. But suppose we set the criteria for predicting high value at 40%. Then these observations would be predicted to be high value. This leads to tradeoffs as predicting more high value customers will lead to more errors on those predictions, but fewer errors when predicting low value. The ROC curve shows the sensitivity and specificity at various probability cutoffs. Curves that bend to the upper left of the square represent greater accuracy, and hence the area under the curve (AUC) is an overall measure of accuracy.

Task 5 – Consider a random forest (5 points)

Candidates were able to describe similarities of random forests and boosted trees, but many struggled to describe differences.

Random forests use bagging to produce many independent trees. When a split is to be considered, only a randomly selected subset of variables is considered. In addition, each tree is built from a bootstrap sample from the data. This approach is designed to reduce overfitting, which is likely when a single tree is used.

Boosted trees are built sequentially. A first tree is built in the usual way. Then another tree is fit to the residuals (errors) and is added to the first tree. This allows the second tree to focus on observations on

which the first tree performed poorly. Additional trees are then added with cross-validation used to decide when to stop (and prevent overfitting).

Both random forests and boosted trees are ensemble methods, which means that rather than creating a single tree, many trees are produced. Neither method is transparent, so they require variable importance measures to understand the extent to which input variable is used.

Random forests do not use the residuals to build successive trees, so there is less risk of overfitting as a result of building too many trees. Random forests will typically reduce the variance of predictions when compared to boosted trees, while boosted trees will result in a lower bias. The best boosted trees learn slowly (use lots of trees) and thus can take longer than a random forest to train.

Task 6 – Convert cap_gain to a factor variable (5 points)

Most candidates were able to use the decision tree from task 2 to justify break points, but many candidates struggled to describe potential benefits of this grouping. Candidates could earn full points using splits not explicitly stated in the tree such as creating a category for zeros or rounding the break point values, provided a justification was provided.

The constructed tree split the cap_gain variable at 5095.5 and 7055.5. I created a new variable, cap_gain_cut with values lowcg for those below 5095.5, mediumcg for those between 5095.5 and 7055.5, and highcg for those above 7055.5. The new grouping is shown in the following table.

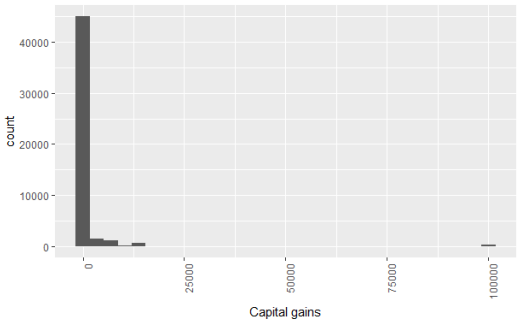
cap_gain_cut	count
lowcg	38,110
mediumcg	276
highcg	2,024

The constructed tree split the cap_gain variable at 5095.5 and 7055.5. I created a new variable, cap_gain_cut with values lowcg for those below 5095.5, mediumcg for those between 5095.5 and 7055.5, and highcg for those above 7055.5. The new grouping is shown in the following table.

cap_gain_cut	count
lowcg	38,110
mediumcg	276
highcg	2,024

It turns out that few are in the mediumcg category (276 overall), but I will stick with it given that the tree indicated this was an important division.

This new variable may be more useful in a GLM for two reasons. First, the cap_gain variable is highly skewed, as seen in the following histogram.



Second, the vast majority of observations (36,592) have a value of zero. Skewed variables are often handled by taking logarithms, but this is not possible when there are zeros. An option would be to set an indicator variable for those with a value of zero, but that seems harder to interpret. I considered making a fourth bucket, just for those with a value of zero, but again decided to go with the tree results.

This approach will be easy to interpret when it comes to model building (understanding that interpretation is not the primary goal) and not require artificial adjustments.

Task 7 – Select a distribution and link function for a GLM (3 points)

Most candidates made appropriate choices for their distribution and link function, but many candidates failed to adequately justify their choices.

Because the target variable is zero or one, the natural distribution choice is binomial. None of the other allowed distributions have this property. For the link function it is essential that the prediction be in the zero to one range because we are predicting the probability of being high value. There are four available link functions that do this: logit, probit, cauchit, and cloglog. All of them can work here and all four are more difficult to interpret than the simpler log or identity links. Given that interpretation is not the primary goal, I would be comfortable with any of them. The logit link is the canonical link, which results in faster processing and more likely convergence. So I will use that link function.

Task 8 – Estimate the GLM coefficients using regularization (10 points)

Most candidates successfully used R to create a GLM and implement regularized regression. Many candidates failed to adequately compare the usage of the variables in the GLM to the earlier tree based models. Graders assessed each grader's comparisons based on the unique variable decisions made in the earlier tasks. In order to earn full points, candidates needed to go beyond listing the variables used by discussing why they were used by the different algorithms. Candidates needed to include evidence of running Ridge, LASSO, and elsastic net in their report to earn full points.

The glmnet package for conducting a regularized regression allows for the binomial distribution and logit link, so that combination will be used. The interaction between marital status and education has been added to the model.

There are two parameters to select when performing regularization with elasticnet. The alpha parameter selects ridge regression, the lasso, or a combination. The lambda parameter controls the regularization penalty. For a given alpha, the program selects the lambda that minimizes cross-validation error on the training set. Then a model fit with that lambda can be fit and evaluated against the testing set.

I tested three alpha values. At alpha = 1 (lasso) the AUC is 0.8870, at alpha = 0 (ridge) the AUC is 0.8863, and at alpha = 0.5 (elasticnet) the AUC is 0.887. Given the similarity of the AUC values I did not see a need to test additional alpha values. With the nearly identical values, I will choose the lasso as it has the advantage of potentially eliminating variables.

This model has an intercept of -9.603832. The coefficients are:

age	0.018454
education_num	0.272148
marital_statusDivorced	-2.18302
marital_statusMarried-spouse-absent	-0.95211
marital_statusNever-married	-2.48073
marital_statusSeparated	-2.31158
marital_statusWidowed	-1.7342
occupationGroup 2	0.510657
occupationGroup 3	0.846805
occupationGroup 4	1.253748
occupationGroup 5	1.43366
occupationGroup NA	0.153594
hours_per_week	0.021848
score	0.061812
cap_gain_cutmediumcg	1.866129
cap_gain_cuthighcg	5.41113
education_num:marital_statusDivorced	.
education_num:marital_statusMarried-spouse-absent	-0.09666
education_num:marital_statusNever-married	.
education_num:marital_statusSeparated	.
education_num:marital_statusWidowed	-0.0365

We see that three of the interactions were not used in the final model. While the regularization kept two of the five interaction levels, most have small coefficients and it may be appropriate to drop the interaction altogether.

The Task 2 tree did not use score while both the regularized regression and the boosted tree did use score. The treatment of continuous variables in GLMs and single decision trees is very different. To effectively use a continuous variable, trees need to repeatedly split to capture the effect, whereas the GLM requires only a single coefficient to be estimated. Because our trees had limited depth, it is not surprising that score was not used effectively. Using the residuals to build successive trees, the boosted tree algorithm was able to find importance in the score variable.

The boosted tree found no importance in the NA occupation group, which was used in the GLM and the Task 2 tree.

Some candidates stated that binarization leads to a more complex model. This is not true. Complexity in a GLM relates to the number of parameters to be estimated. Whether binarized or not, a factor variable requires the same number of parameters.

Binarization allows algorithms to drop individual factor levels if they are not significant as compared to the base level. This can lead to a simpler model with fewer coefficients to estimate. The disadvantage of binarization is that nonsensical results can be obtained if only a handful of factor levels are included in the model. For example, if I binarized education_num, I might find that having education_num = 7 leads to higher value applicants and is the only factor level included in the model, suggesting that education only matters if you stop at that exact education_num.

Task 9 – Select the final model (5 points)

Most candidates demonstrated an understanding of the items to consider when selecting a model and such as performance and interpretability. Many candidates failed to point out how their selected model met the business needs for this problem.

The GLMs coefficients can yield a clear understanding of variable effects, whereas one must rely on variable importance measures to gain insights about the important variables. Furthermore, a GLM could be implemented with a spreadsheet, which would be far easier than the boosted tree implementation. One advantage of the boosted tree is that it can automatically capture the effects of variable interactions. Because successive trees fit based on the residuals, boosted trees can lead to a superior fit. In our case, the ability of our model to predict which customers will be high value is the most important factor.

For our models, the AUC can be interpreted as the proportion of high value applicants ranked ahead of a random low value applicant. The Task 4 boosted tree had an AUC of 0.8991 on the test set while the GLM fit by lasso had 0.8870. This difference is not much, though slightly favors the boosted tree. Given that prediction is the main goal, I recommend using the boosted tree as my final model. I also note that with more time I might be able to tune the tree to improve its accuracy. There isn't much more that can be done with the GLM.

Task 10 – Select the cutoff probability (10 points)

Most candidates were able to calculate the expected profit correctly, but many were not able to write a clear communication for the marketing audience. To receive full points, there needed to be evidence of the process used to determine the cutoff; typically this was done with code comments or documentation of results in the report. Some candidates struggled to explain results and make appropriate decisions when cutoffs resulted in ties. This was particularly true when the selected model was a simple decision tree that yielded a small number of possible probabilities.

The algorithm we have produced takes the characteristics of a policyholder and puts them through a formula that provides the probability that policyholder will be a high-value customer. To implement the model, a cutoff probability between 0 and 1 must be chosen. All applicants who score above the cutoff will be predicted to be a high-value customer to whom we would market a policy. All applicants who score below the cutoff will be predicted to be a low-value customer to whom we would not market a policy. If we set the cutoff too high we will avoid selling to many low-value applicants, but we won't market to enough high-value applicants. If we set the cutoff too low, we will market to too many low-

value customers. Because there is more upside to correctly identifying a high-value customer we want to err on the side of predicting more high-value customers.

Using trial and error, I selected the cutoff that maximizes profits. The table below shows the process including the cutoffs chosen and resulting profit.

Trial Number	Cutoff	Profit
1	0.5	43,305
2	0.4	54,660
3	0.3	63,240 (Best)
4	0.2	62,005
5	0.25	62,650
6	0.28	63,050
7	0.29	63,215
8	0.31	62,195

The best cutoff to two decimal places is 0.30 with an expected profit of 63,240. Applying that cutoff to our 12,123 sample policies, here is what we see:

- 4,566 are predicted to be high value and are selected for marketing. Of them,
  - 1,697 applicants are actually low value. We lose 25 on each for a total loss of 42,425.
  - 2,869 are actually high value. We gain 50 on each for a total gain of 143,450.
- 7,557 are predicted to be of low value and are not selected for marketing. We lose 5 on each for a total loss of 37,785.

The total expected profit is 143,450 – 42,425 – 37,785 = 63,240.

Task 11 – Write model demo for marketing (10 points)

Many candidates struggled with this task. Candidates needed to include sample cases that resulted in low and high value predictions and clearly describe the analysis for the marketing team. Many candidates who selected a decision tree model attempted to walk through how the cases would flow through the tree diagram, which proved more difficult than just discussing the directional effects based on output probabilities. Candidates were encouraged to modify the supplied cases. Few elected to test changes in both directions from the base case.

To predict if a potential customer will be high or low value we will use seven pieces of information. In the table below I have put eight fictitious applicants through the model and in each case calculated the probability that this individual is a high-value customer. Based on a cost-benefit analysis, applicants with a probability of 0.25 or less will be rated as low value. Such applicants are indicated in the final column of the table below. The characteristic that is changing relative to the first row is indicated in bold type.

age	education_num	marital_status	occupation	cap_gain	hours_per_week	score	Prob of high	Value
39	10	Married-spouse	Group 3	0	40	60	0.378	High
53	10	Married-spouse	Group 3	0	40	60	0.457	High
25	10	Married-spouse	Group 3	0	40	60	0.203	Low
39	**13**	Married-spouse	Group 3	0	40	60	0.552	High

39	**7**	Married-spouse	Group 3	0	40	60	0.120	Low
39	10	**Never-married**	Group 3	0	40	60	0.036	Low
39	10	Married-spouse	**Group 5**	0	40	60	0.569	High
39	10	Married-spouse	**Group 1**	0	40	60	0.294	Low
39	10	Married-spouse	Group 3	**6000**	40	60	0.693	High
39	10	Married-spouse	Group 3	0	**50**	60	0.447	High
39	10	Married-spouse	Group 3	0	**20**	60	0.216	Low
39	10	Married-spouse	Group 3	0	40	**64**	0.417	High
39	10	Married-spouse	Group 3	0	40	**44**	0.252	Low

The first row represents the most common situation. We can see several ways that changing a single customer attribute affects the output probability of being high value. Changing to never married, decreasing age to 25, decreasing education num to 7, changing to occupation Group 1, decreasing hours worked per week to 20, or lowering the score to 44 all resulted in a low-value prediction. Alternatively, increasing age, education num, occupation group number, capital gains, hours worked per week, or score all increased the probability of being high value. However, it is important to note that these results indicate direction of changing a single attribute, but the probability depends on all the inputs.

Task 12 – Executive summary (20 points)

Most candidates successfully described the data and the business problem. Some candidates failed to describe the quantity of data or the nature of the variables. Some candidates misinterpreted the business problem and thought it was to identify the variables that were important. Many candidates did not clearly state the model they selected, adequately justify their model recommendation, or describe the impact to profitability of implementation. The best candidates used language that was free from unexplained technical jargon.

From: Actuarial Analyst

You have asked us to build a model that can predict if an applicant will be a high-value or low-value customer. We were supplied with 48,842 observations on prior customers that can be used to build a model and gauge the potential profit from using it. I was informed that this product would be sold only to customers age 25 and older, so younger policyholders were removed from the data, leaving 40,410 to analyze. The model we constructed is relatively simple to apply and produced a projected profit of 5.22 per applicant.

Prior to building the model we checked the data in a variety of ways. The data contained the applicant's age, education level, marital status, occupation group, capital gains, the number of hours worked per week, a proprietary insurance score developed by MEB, and an indicator of whether the applicant was high or low value. Other than for occupation group, there were no missing values.

In examining the data it appears that all the variables may be useful in predicting customer value. In particular, high value tends to be associated with older ages, more years of education, being married with spouse present, being in a high-numbered occupation group, working more hours per week, having a higher insurance score, and having more capital gains. I also noted that there were two types of

married individuals. One had very few observations and a similar relationship to being high value. Those two were combined into a single category of being married with spouse present.

I then tried a variety of models to see which would perform best. This is done by calibrating a model on 70% of the data and then seeing how that model performs on the other 30%. This replicates the way our model will be used in that the model will be predicting value for new applicants, who were not used to build the model. Regardless of model type, the output for a given applicant is the probability that applicant will be high value. The higher the probability, the more likely it is that the applicant will generate profit for the company. As part of the modeling process I then need to determine the cutoff, the point where the probability is high enough to decide that marketing to this individual will be profitable.

The selected boosted tree model was more capable of ranking high-value customers ahead of low-value customers, as measured by AUC – a common measure for comparing models. Further analysis showed that we should consider any customer with probability 30% or more as potentially high value. This is because the rewards of finding high-value customers exceed the costs of incorrectly marketing to low-value customers. The following were important inputs to the model:

- Amount of capital gains
- Education level
- Marital status
- Age
- Insurance score
- Hours worked per week
- Occupation group

The impact of each of these variables in the model is complex, so I performed sensitivity analysis to understand the impacts. As an illustration of the model effects and selected cutoff, consider a baseline customer with the following characteristics:

- 39 years old
- Education level of 10
- Married with spouse present
- In occupation group 3
- No capital gains
- Works 40 hours per week
- Insurance score of 60

My model assigns a probability of 38% that this applicant will be of high value. Because that probability exceeds our 30% cutoff, we would predict the applicant is high value and market a policy to them. If the same applicant had never been married, the customer would only have a 4% probability of being high value. Therefore, we would predict low value and avoid marketing a policy to them. Conversely, if the original customer had 6,000 in capital gains, the probability of being high value jumps to 69%. It makes sense that older customers with more education, stable marriages, and investment earnings would be higher value applicants.

We tested the model and cutoff on 12,123 customers that were not used to build the model. Out of those applicants, only 38% would have been selected for marketing of which 63% would be high value. This results in a net profit of 63,240, equivalent to 5.22 per applicant. As an alternative, consider the outcome if we marketed to all 12,123 applicants. Only 29% of those would be high value, resulting in a net loss of 41,250. It is clear that the model and cutoff are effective at targeting a higher value pool of applicants, which could lead to high profits.

We are confident that this model will improve profitability and recommend implementation.