DATA MINING NOTES

BRENDAN WHITAKER

2010 *Mathematics Subject Classification.* 12-XX

Abstract. A comprehensive set of notes for Data Mining course, taken SP18 at The Ohio State University.

Contents

Part 1

CHAPTER 1

Introduction

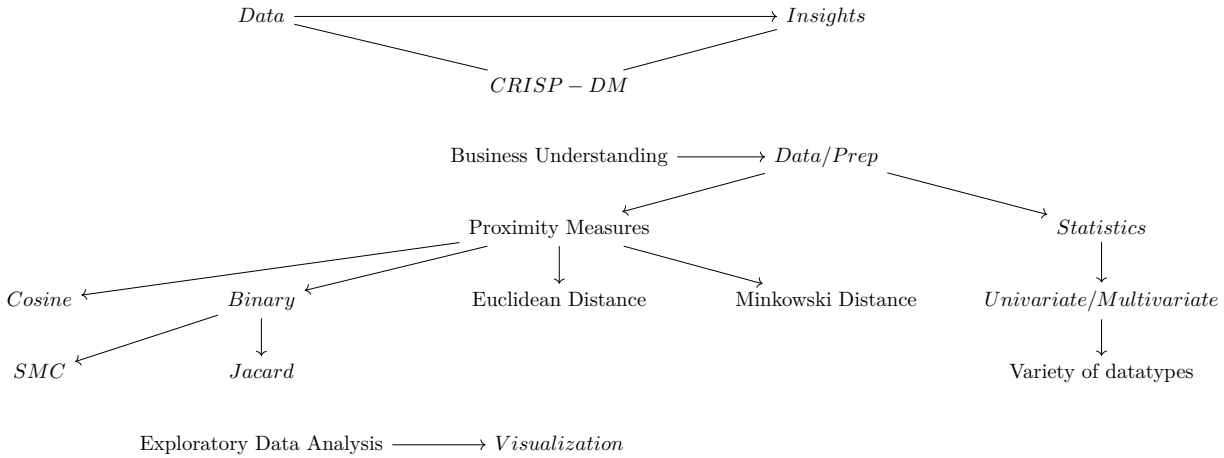
Wednesday, January 10th

Note $class = label = category$. These are the dependent variables, the stuff that our work is determining.

Ratio variables are your typical real-valued numbers, i.e. zero is meaningful. Interval variables have 0 as just another possible value, it is not the additive identity in this case.

Monday, January 22nd

Review of stuff up to now.



Imputation: replacing missing data with the mean is one way we could do it. We could also use regression imputation to estimate the value of the missing variable using the data from other variables. You could also use a random value.

Monday, January 29th

PCA - principal component analysis.

Monday, Febrary 5th We have:

$$P(+|x) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2}}.$$

Okay so last time, we talked about how you might compute a ROC curve. Recall that a ROC curve plots false positive rate (FPR) on the x-axis and true positive rate (TPR) on the y-axis. So we are going to set our $t = 0$.

1.1. Exam 1

Need to know proximity measures, types of data (ratio).

- Won't ask to compute covariance, that's mean.
- Know bayes theorem for bayes classifiers.
- PROXIMITY MEASURES, INFER IF WE'RE TALKING ABOUT A SIMILARITY VS DISSIMILARITY.
- know minkowski distance.
- don't know mahalaboni distance.
- simple matching and jaccard coefficients, and cosine.
- you use jacard when data is SPARSE, otherwise SMC.
- Make a reference sheet to use on exam.
- You'd use cosine when it is no longer binary data, but it can be used for anything.
- SMC and Jaccard are for BINARY ONLY.
- Know that the end of boxplots are 90 and 10.
- Interpret this scatterplot or boxplot on exam. Any of the data vis types.
- What is the difference between noise and outliers?
- Principal component analysis is going to be fair game, or part of the calculation, how do you do it, what does it mean, what are you trying to accomplish.
- how many principal components do you keep given some type of data loss requirement.
- Difference between dimensionality reduction, and feature subset selection.

- Dimensionality reduction creates new features, no longer interpretable, the values are meaningless, each of the new dimensions are combinations/transformations of the old ones.
- Feature subset selection just removes a subset of the n dimensions and uses those, which preserves the meaning of the dimensions/features.
- When you run PCA, you aren't doing anything related to the class variable.
- But Feature subset selection on the other hand, is usually done relative to a classifier.

Now let's talk about classification.

- What is objective of classification: create a generalizeable, predictive model. It has to GENERALIZE!
- As you train your model and it becomes more complex, training error keeps dropping, but test error find a local min and then goes up again.
- Producing a classification label is a two step process.

EXAMPLE 1.1. Suppose we're doing a binary classification.
Step 1:
For a given record x, define:

$$P(+|x) = 0.64 \tag{1.1}$$
$$P(-|x) = 0.36$$

For logistic regression, we have:

$$P(+|x) = \frac{1}{1 + e^{-*\beta_0 + \beta_1 x)}}.$$

So x enters our model and get the the above posterior out of it.
Step 2:
We set some threshold t and say $x \in +$ if the posterior is $\geq t$ and in $-$ otherwise. And 0.5 is our default t value. You get an ROC curve only by varying t. Note the confusion matrix is dependent on the value of t.

- A ROC curve has FPR on x axis and TPR on y axis.
- When we increase t, precision generally goes up because we expect the model to work better than random.
- Recall precision is TP/(TP + FP), it is the top left box over the sum of the left half of the matrix.
- You should know DECISION TREES, NAIVE BAYES, kNN. Ensembles just a little bit. Rule based classifier.
- **What is reduced-error pruning?** Before we talk about that, normal pruning is just computing error for all nodes and removing nodes that don't give you lift (reduction of error rate). Reduced error pruning using validation data, you compute error rates for each node on the TEST DATA instead, then compute lift and decide to prune any nodes that don't give you a reduction in error.
- for notes sheet just print a shitton of these pages.

1.2. Transcript

Jason Van Hulse Data Mining Introduction Spring 2018 1 Data Mining: Why? Commercial Viewpoint

- Lots of data is being collected and warehoused
- Web, image, voice, video, text data
- purchases at department/grocery stores
- Bank/Credit Card transactions
- Internet of Things
- Computers have become cheaper and more powerful
- Competitive Pressure is Strong
- Provide a competitive advantage in the marketplace
- Creation of data products 2 Scientific Viewpoint :
- Data collected and stored at enormous speeds (GB+ /hour)
- remote sensors on a satellite
- telescopes scanning the skies
- microarrays generating gene expression data
- scientific simulations
- Traditional techniques infeasible for raw data
- Data mining may help scientists
- in classifying and segmenting data
- in hypothesis f ormation Data Mining: Why?
- There is often information "hidden" in the data that is not readily evident
- Human analysts may take weeks to discover useful information
- Much of the data is never analyzed at all 3 What is Data Mining?
- Data mining is the non-trivial extraction of implicit , previously unknown and potentially useful information from data

- Data mining is a technology that blends data analysis methods with sophisticated algorithms for processing large volumes of data.
- Data mining is a key step in the knowledge discovery process . Data contains value and knowledge! But to extract knowledge, data must be
- Stored
- Managed
- Analyzed 4 The Origins of Data Mining
- Data mining draws ideas from statistics, artificial intelligence, machine learning, and data systems.
- Traditional techniques may be unsuitable due to
- Enormity of data
- High dimensionality of data
- Heterogeneous, distributed nature of data Artificial Intelligence/Machine Learning Statistic s Data Mining Data systems 5 The Origins of Data Mining
- Data Mining borrows different concepts from a number of different domains, and is very cross-disciplinary in approach:
- Data Mining is a type of induction , e.g., proceeding from very specific knowledge to a more general concept
- Data Mining is a type of compression since it allows detailed data to be abstracted or summarized in an interesting and meaningful way
- Data Mining can be viewed as a type of complex query of a database
- Data Mining can describe a large set of data by an approximation
- Data mining can be thought of as a type of search problem - finding some structure in a large and complex set of data 6 Data Mining Tasks
- Prediction Methods : Use some variables to predict unknown or future values of other variables.
- Classification
- Regression
- Deviation Detection 7
- Description Methods : Find human- interpretable patterns that describe the data
- Clustering
- Association Rule Discovery
- Sequential Pattern Discovery

Types of attributes: We have **nominal**, which only has distinctness ($=, \neq$). We have **ordinal**, which has distinctness and order ($<>$) (letter grades), we have **interval**, which has distinct, order, addition (temp in celsius), and we have **ratio**, which has all including multiplication (age).
$s_{xy} = Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}[(x_i - \overline{x})(y_i - \overline{y})]$. $s_x = StdDev(x) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$. $Corr(X,Y) = \frac{s_{xy}}{s_x s_y}$. Note SMC and Jaccard Coefficient are for Binary data only. $J(x,y) = \frac{f_{11}}{f_{01}f_{10}f_{11}}$. f_{11} is the number of entries which are nonzero in both x, y and f_{10} is entries nonzero in x only. We discuss similarity measures for 4 data types. Similarity is [0,1], and dissimilarity, min is 0, max varies. For nominal use binary for sim and 1- binary for dissim. FOr ordinal use dissim $d = \frac{|p-q|}{n-1}$, where n is the number of values. For sim use $1-d$. For interval/ratio use dissim $d = |p-q|$ and sim is $s = \frac{1}{1-d}$ or $-d$ or $\frac{d-min_d}{max_d - min_d}$. **Minkowski:** $dist = \left(\sum_{k=1}^{n}|p_k - q_k|^r\right)^{1/r}$. $r = 1$ is city block distance. $r = 2$ is euclidean. Distances must be ≥ 0 for all and 0 iff $p = q$ pos def, symmetric, and triangle ineq all three gives us a metric. **SMC** simple matching is number of matches over number of attributes (binary). Ends of boxplot are 10 and 90 percent. Use jaccard for sparse, use euclidean for cts dense. **Representation** is how data is mapped to visual format. **Selection** is elimination or demphasis of certain objects atrobutes. **Arrangement** is placing of visual eleemnts in display. **Noise** refers to modification of original values (distorion of voice on phone, static on screeen) (measurement error, flaws in data colelction). **Outliers are data objects with characterisitcs that are vastly different from most of data in set. Curse of dimensionality** num dimensions/attributes increases volume increases so fast that everything gets hella sparse. Types of dimensionality reduction: **PCA** find a lower dimensional representation that captures the largest amount of variation in the data, does not preserve meaning of axes. **INSERT PICTURE FROM SLIDES**. (Find new basis by diagonalization). **Feature subset selection** you just cut off axes that aren't that valuable, preserves the meaning of remaining axes. Brute force it by trying every possible combination of removal, embedded approach does it naturally as part of data mining algo (stepwise regression) and filter appraoches filters before mining begins. When you run PCA, you aren't doing anything related to the class variable, but in feature subset selection, you're doing it relative to the class variable. **Bayes Theorem** $P(Y|X) = \frac{P(X|Y)}{P(X)}P(Y)$ the fraction is called the support, if support is greater than 1, then observed data X will increase your belief in Y.
CLASSIFICATION
Objective of classficiation is to find a **GENERALIZEABLE** predictive model which does better than random selection. We have the graph that he drew on the board where as you train your model and the model becomes more complex, the training error rate continues to drop off, but at a certain point, the test error rate finds a local min and then begins to rise again, so the optimal model is at this local min. Producing a classification label is a two step process: **STEP 1:** Suppose we're given a record x and we're going to do logistic regression on it. Then our equation is $P(+|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$. So we get the posteriors $P(+|x) = 0.64, P(-|x) = 0.36$. Now we have **STEP 2:** We set some threshold t and say $x \in +$ if the posterior is $\geq t$ and in $-$ otherwise. And 0.5 is our default t. You get a ROC curve by only varying t and plotting FPR, TRP. Note the confusion matrix is dependent on t. A **ROC CURVE** has FPR on x axis and TPR on y-axis. When we increase t, precision generally goes up because we expect the model to work better than random. Recall **precision** is $\frac{TP}{TP+FP}$. it is the top left box over the sum of the left half of the matrix. **linear regression** is when we try to fit a linear model to the observed (usually cts) data. Equation is $f(x) = w_1 x + w_0$. We minimize the sum of squared errors $E = \sum_i (y_i - f(x_i))^2 = \sum_i (y_i - w_1 x_i - w_0))^2$. To actually do it, you take the partial with respect to w_0, w_1 and get two equations set to zero, then solve for the weights. **Total sum of squares** $SST = \sum_i (y_i - mean(y))^2$. **model sum of squares** $SSM = \sum_i (f(x_i) - mean(y))^2$. Total sum of squares can be partitioned into model and error terms $SST = SSM + SSE$. And Goodness of fit $R^2 = SSM/SST$. Multiple linear regression has $x_1, ..., x_n$ and $n+1$ weights and a single target $f(x) = w_0 + w_1 x_1 + \cdots + w_n x_n$. **Logisitc regression** is for binary classification problem. Define $p(x) = P(Y = 1|x)$. then $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$. And $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$. SO the logit of p(x) is linear. For several variables, make the β linear comb longer on top and bottom. **Confusion Matrix**. Predicted class on top, actual class on left. Top left is TP, top right is FN, bottom left is FP, bottom right is TN. **Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$. **Error Rate:** $1 - Accuracy$. TPR (sensitivity) $\frac{TP}{TP+FN}$. TNR (specificity) $= \frac{TN}{TN+FP}$. FPR $= \frac{FP}{FP+TN} = 1 - TNR$. FNR $= \frac{FN}{FN+TP} = 1 - TPR$. **kNN** Add a new point, take majority vote of class labels among k nearest neighbors, or compute the posterior $P(+|x)$ as the proportion of K nearsest neighbors that are positive. You could also weight the vote according to distance $w = \frac{1}{d^2}$. if K is too small sensitive to noise points, if too large, neighborhood may include points form other classes. Lazy learners. **Cost sensitive measures**. **Precision** $= \frac{TP}{TP+FP}$. **Recall** $= \frac{TP}{TP+FN}$.(same as TPR). **F-measure** $= 2(precision)(recall)/(precision + recall) = \frac{2TP}{2TP+FN+FP}$. We talk about **ROC curve**. Lift is improvement over random class. 0,0 declares everything to be negative, 1,1 declares evryting positive, 0,1 is ideal perfect. Just compute TPR and FPR for each threshold t value and graph it. easy. **Resubstitution errors** $\sum e(t)$ error on training data. **Generalization errors** $\sum e'(t)$ error on test data. Optimistic approach assumes $e(t) = e'(t)$. So assume test error is training error. Occams razor is prefer simpler models all else equal. Single holdout is reserve 2/3 for training rest for test. Random subsampling is repeat singleholdout k times. Cross validation is partition data into K subsets and train on $k-1$ partitions, test on remaining. Average the results. Stratified is class distribition kept equal across each test partition. Leave one out is $k = n$. Bootstrap is sampling with replacement to obtain distinct datasets, N observation bootstrap of size N has on average 63.2 percent of observations in it. Overfitting is when model is too complex, doesn't generalize. Underfitting doesnt capture enough complexity. Homogeneous nodes are better, mostly all one class. $GINI(t) = 1 - \sum_i [p(j|t)]^2$, where $p(j|t)$ is the relative frequency of class j at node t. It has a maximum of $1 - \frac{1}{n_c}$ when records equally distributed among classes, min of 0, when all 1 class, most interesting. When a node p is split into k partitions, children, the quality of split is $GINI_{split} = \sum_{i=1}^{k}\frac{n_i}{n}GINI(i)$. where n_i is number of records at child i and n is number of records at node p. Use gini to choose best two way split. **Entropy** at node t is $ENTROPY(t) = -\sum_j p(j|t)\log_2 p(j|t)$. **Info gain** is $GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k}\frac{n_i}{n}Entropy(i)\right)$. n_i is records in node i. Want to maximize gain. Disadvantage is tends to prefer small but pure splits. So we use $GAINRATIO_{split} = \frac{GAIN_{split}}{SPLITINFO}$ where $SPLITINFO = -\sum_{i=1}^{k}\frac{n_i}{n}\log_2\frac{n_i}{n}$. Classification error at node t is $Error(t) = 1 - \max P(i|t)$. **When to stop splitting.** Pre-pruning is the early stopping rule, stop before it becomes fully grown tree. 1. Stop if all instances belong to same class, 2. stop if all attribute values are the same. **More restrictive conditions**: Stop if number of instances is less than some threshold. Stop if class distribution of instances are independent of available features using χ^2 test. Stop if expanding the current node does not improve impurity measures (Gini or info gain). **Post pruning.** Grow tree to it's entirety. Trim nodes of decision tree bottom up. If generalization error improves after trimming, replace subtree by a leaf node. Class label of leaf node is determined by majority class of instances in subtree. Reduced Error pruning, uses a holdout dataset to estimate genrealization error. Pessimistic error rate is total errors $e'(T) = e(T) + N(0.5)$ where N is number of leaf nodes. **Naive bayes** all effects are independent given common cause (class). So attributes indep given class. **Rule based classifier** uses a collection of if then statements. (atomic logic). Rule r covers instance x if it satisfies conditions of rule. Accuracy of rule is fraction of instances that satisfy the condition and resulto frule. **Rule evaluation Accuracy** $= \frac{n_+}{n}$ and Laplace is $= \frac{n_+ + 1}{n+k}$ and $M - estimate$ is $= \frac{n_+ + kp}{n+k}$. n is number of instances covered by rule, n_+ is number of positive instances covered by rule, k is number of classes, and p is prior prob. Ensemble classfiers are effective when the aggregate many DIFFERENT classifiers. They must be independent.

(1) *We compute the Gini index of overall collection.* Let t_0 be the root node of the entire collection of training examples. Observe: $Gini(t_0) = 1 - \sum_{i=0}^{1}[p(i|t_0)]^2 = 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 1 - \left(\frac{1}{4} + \frac{1}{4}\right) = 1 - \frac{1}{2} = \frac{1}{2}$.

(2) *We compute the Gini index of* `Customer ID`.
Let t_j be the j-th `Customer ID` node. Since each node only has a single data point, the sum will be equal to 1, since one of the probabilities $p(i|t_j)$ will be 1, and the other 0. So $Gini(t_j) = 0$ for all j. Hence the Gini index is 0.

(3) *We compute the Gini index of* `gender`.
Let t_M be the male node and t_F be the female node. We have: $Gini(t_M) = 1 - \sum_{i=0}^{1}[p(i|t_M)]^2 = 1 - \left(\left(\frac{4}{10}\right)^2 + \left(\frac{6}{10}\right)^2\right) = 1 - \left(\frac{4}{25} + \frac{9}{25}\right) = 1 - \frac{13}{25} = \frac{12}{25}$.$Gini(t_F) = 1 - \sum_{i=0}^{1}[p(i|t_F)]^2 = 1 - \left(\left(\frac{4}{10}\right)^2 + \left(\frac{6}{10}\right)^2\right) = 1 - \left(\frac{4}{25} + \frac{9}{25}\right) = 1 - \frac{13}{25} = \frac{12}{25}$. So the Gini index is $\frac{10}{20}\frac{12}{25} + \frac{10}{20}\frac{12}{25} = \frac{12}{25} = 0.48$.

(4) *We compute the Gini index of* `car type`.
Let t_F be the family node and t_S be the sports node, and let t_L be the luxury node. We have: $Gini(t_S) = 1 - \sum_{i=0}^{1}[p(i|t_S)]^2 = 1 - \left(\left(\frac{10}{10}\right)^2 + \left(\frac{0}{10}\right)^2\right) = 1 - 1 = 0.Gini(t_F) = 1 - \sum_{i=0}^{1}[p(i|t_F)]^2 = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = 1 - \left(\frac{1}{16} + \frac{9}{16}\right) = 1 - \frac{5}{8} = \frac{3}{8}.Gini(t_L) = 1 - \sum_{i=0}^{1}[p(i|t_L)]^2 = 1 - \left(\left(\frac{1}{8}\right)^2 + \left(\frac{7}{8}\right)^2\right) = 1 - \left(\frac{1}{64} + \frac{49}{64}\right) = 1 - \frac{25}{32} = \frac{7}{32}$. So the Gini index is $\frac{8}{20}0 + \frac{4}{20}\frac{3}{8} + \frac{8}{20}\frac{7}{32} = 0.1625$.

(5) *We compute the Gini index of* `shirt size`.
Let t_S be the small node and t_M be the medium node, and let t_L be the large node, and t_E be the extra large node. We have: $Gini(t_S) = 1 - \sum_{i=0}^{1}[p(i|t_S)]^2 = 1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right) = 1 - \left(\frac{9}{25} + \frac{4}{25}\right) = 1 - \frac{13}{25} = \frac{12}{25}.Gini(t_M) = 1 - \sum_{i=0}^{1}[p(i|t_M)]^2 = 1 - \left(\left(\frac{3}{7}\right)^2 + \left(\frac{4}{7}\right)^2\right) = 1 - \left(\frac{9}{49} + \frac{16}{49}\right) = 1 - \frac{25}{49} = \frac{24}{49}.Gini(t_L) = 1 - \sum_{i=0}^{1}[p(i|t_L)]^2 = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 1 - \left(\frac{1}{4} + \frac{1}{4}\right) = 1 - \frac{1}{2} = \frac{1}{2}.Gini(t_E) = 1 - \sum_{i=0}^{1}[p(i|t_E)]^2 = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 1 - \left(\frac{1}{4} + \frac{1}{4}\right) = 1 - \frac{1}{2} = \frac{1}{2}$. So the Gini index is $\frac{5}{20}\frac{12}{25} + \frac{7}{20}\frac{24}{49} + \frac{4}{20}\frac{1}{2} + \frac{4}{20}\frac{1}{2} = 0.3914$.

(6) *Which of these 3 is preferred?*
Car type is preferred since its Gini index is lowest at 0.1625.

(7) *Explain why* `customer ID` *should not be used even though it's genie index is zero.*
`customer ID` should not be used since it only has a single data point per node, so it's likely that the customer ID's in any test data will not be ones we have already seen, assuming customer ID is unique, so we won't be able to use our model to gain any new information about the test set.

(1) *Find entropy of examples with respect to positive class.* Observe: $Entropy = -\sum_{i=0}^{1} p(i)\log_2 p(i) = -\left(\frac{4}{9}\log_2\frac{4}{9} + \frac{5}{9}\log_2\frac{5}{9}\right) = -(-.52 - .471) = 0.991.$

(2) *What are the information gains of a_1 and a_2 relative to these training examples?* Let I be entropy. Recall: $\Delta_{info} = I - \sum_{j=1}^{k}\frac{N(v_j)}{N}I(v_j)$ So we have: $\Delta_{info} = 0.991 - \sum_{j=1}^{2}\frac{N(a_i)}{9}I(a_i)$ We compute: $I(a_1 = T) = -\sum_{i=0}^{1} p(i|a_1 = F)\log_2 p(i|a_1 = F) = -(-.5 - .311) = 0.811. I(a_1 = F) = -\sum_{i=0}^{1} p(i|a_1 = F)\log_2 p(i|a_1 = F) = 0.971 I(a_2 = T) = -\sum_{i=0}^{1} p(i|a_2 = T)\log_2 p(i|a_2 = T) = -(-.529 - .442) = 0.971 I(a_2 = F) = -\sum_{i=0}^{1} p(i|a_2 = F)\log_2 p(i|a_2 = F) = 1.$ Then: $\Delta_{info}(a_1) = 0.991 - \sum_{j=1}^{2}\frac{N(a_i)}{9}I(a_i) = 0.991 - \left(\frac{4}{9}0.811 + \frac{5}{9}0.971\right) = 0.091. \Delta_{info}(a_2) = 0.991 - \sum_{j=1}^{2}\frac{N(a_i)}{9}I(a_i) = 0.991 - \left(\frac{4}{9}1 + \frac{5}{9}0.971\right) = 0.016.$

(3) *We compute info gain for every possible split of a_3.*

Let Δ_i denote the info gain of the split $\leq i$ and $> i$. Then we have: $\Delta_{1.5} = 0.991 \Delta_2 = 0.991 - \left(\frac{1}{9}0 + \frac{8}{9}(.954)\right) = 0.143 \Delta_{3.5} = 0.991 - \left(\frac{2}{9} + \frac{7}{9}(.985)\right) = 0.0025 \Delta_{4.5} = 0.991 - \left(\frac{3}{9}(.918) + \frac{6}{9}(.918)\right) = 0.073 \Delta_{5.5} = 0.991 - \left(\frac{5}{9}(.971) + \frac{4}{9}\right) = 0.0071 \Delta_{6.5} = 0.991 - \left(\frac{6}{9} + \frac{3}{9}(.918)\right) = 0.018 \Delta_{7.5} = 0.991 - \left(\frac{8}{9}0 + \frac{1}{9}0\right) = 0.991 \Delta_{8.5} = 0.991.$

(4) The best split of all three is Δ_2 on a_3 because it is non trivial, and it has the highest information gain.

(5) a_1 is better out of first two since it has a lower classification rate.

(6) Gini of T for a_1 is 3/8. Gini of F for a_1 is 8/25. So $Gini(a_1) = 4/9 * 3/8 + 5/9 * 8/25 = 0.344$.

Gini of T for a_2 is 12/25, and for F is 1/2. So $Gini(a_2) = 5/9 * 12/25 + 4/9 * 1/2 = 0.489$. And so a_1 is better again.