Smith School of Business

MMAI 863, AI & ML Technology, Prof. Stephen Thomas

Individual Assignment #1

Blair Nicolle (ID# 201904169)

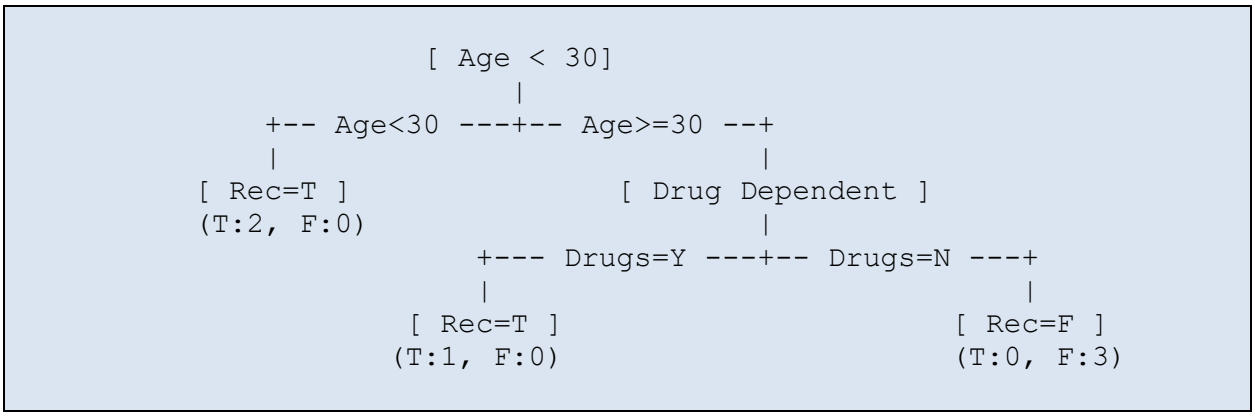
Table of Contents

Question 1 (Jewelry)

For Q#1, please see rendered Jupyter Notebook for in-line responses and Analysis & Discussion at its bottom. For subsequent questions (Q#2-Q#7), I used this doc. for primary assignment responses.

Question 2 (Predicting Recidivists with Decision Trees)

The completed tree appears thus:



Full Solution

Background: The "ID3" algorithm uses entropy-based information gain to construct a decision tree.

Greedy Algorithm

ID3 is called a ***greedy algorithm:*** so, after it makes a split, it doesn't look back based on future splits to decide if it needs to amend the original split in some way to result in a better overall tree. The splits themselves are done in ID3 using information gain calculations.

Formulas

Entropy is defined as:	Information Gain is defined as:
$$E(S) = \sum_{i=1}^{n} - p_i \log_2 p_i$$	Gain(S,A) = E(S) – sum[P(S|A) * E(S|A)], all A The attribute with the largest information gain, G(S,A) will be used to split the tree at each node. The resultant subtree will have reduced entropy.

E(S) ranges from 0 to 1, with 0 being purity and 1 being mixed up in that there is an equality of representation across all classes.	

Source: Smith MMAI -869 class notes, and https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/.

Original Dataset

Behavior	Young	Drugs	Recidivist
F	T	F	T
F	F	F	F
F	T	F	T
T	F	F	F
T	F	T	T
T	F	F	F

We want to predict for:

F	F	T	?????

First Split (Root Node)

The Entropy with respect to the target, i.e. E(Recidivity), of the initial set is 1.0 as it's 50% True and 50% False (a pure mix).

<table><tr><th>Behavior</th><th>Young</th><th>Drugs</th><th>Recidivist</th></tr><tr><td>F</td><td>T</td><td>F</td><td>**T**</td></tr><tr><td>F</td><td>F</td><td>F</td><td>F</td></tr><tr><td>F</td><td>T</td><td>F</td><td>**T**</td></tr></table> P(Behavior=F) = 3/6 = 0.5	E(Recidivist=T, Behavior=F) =E(2,1) = E(.667,.333) = -.667*log(.667) - .333*log(.333) = 0.918296
<table><tr><th>Behavior</th><th>Young</th><th>Drugs</th><th>Recidivist</th></tr><tr><td>T</td><td>F</td><td>F</td><td>F</td></tr><tr><td>T</td><td>F</td><td>T</td><td>**T**</td></tr><tr><td>T</td><td>F</td><td>F</td><td>F</td></tr></table> P(Behavior=T) = 3/6 = 0.5	E(Recidivist=T, Behavior = T) = E (1,2) = E(.333, .667) = -.333*log(.333) - .667*log(.667) = 0.918296
Entropy(Recidivist, Behavior) = P(F)*E(Recidivist, F) + P(T)*E(Recidivist, T) = .5*(0.918296) + .5*(0.918296)	

= 0.918296

InfoGain(Rec,Beh) = E(Rec) – E(Rec,Beh) = 1.00 – 0.918296 = 0.0817

Behavior	Young	Drugs	Recidivist
F	F	F	F
T	F	F	F
T	F	T	**T**
T	F	F	F

P(Young=F) = 4/6 = 0.667

E(Recidivist=T, Young=F)
= E(1,3) = E(.25,.75)
=-.25*log(.25) - .75*log(.75)
= 0.811278

Behavior	Young	Drugs	Recidivist
F	T	F	**T**
F	T	F	**T**

P(Young=T) = 2/6 = 0.333

E(Recidivist=T, Young=T)
=E(2,0) =E(1.0, 0.0)
=-1*log(1) – 0*log(0)
= 0.00

Entropy(Recidivist, Young) = P(F)*E(Recidivist, F) + P(T)*E(Recidivist, T)
= .6667*(0.811278) + .3333*(0.000)
= 0.540852

InfoGain(Rec,Young) = E(Rec) – E(Rec,Young) = 1.00 – 0.540852 = 0.459148

Behavior	Young	Drugs	Recidivist
F	T	F	T
F	F	F	F
F	T	F	T
T	F	F	F
T	F	F	F

P(Drugs=F) = 5/6 = 0.833

E(Recidivist=T, Drug=F)
=E(2,3) =E(.4,.6)
=-.4*log(.4) - .6*log(.6)
= 0.970951

Behavior	Young	Drugs	Recidivist
T	F	T	T

P(Drugs=T) = 1/6 = 0.1667

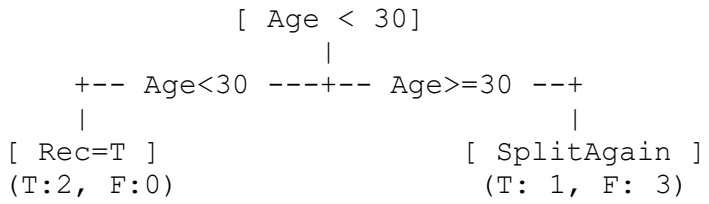
Information Gain = E(Recidivist=T, Drug=T)
=E(1,0) =E(1.0, 0.0)
=-1*log(1) – 0*log(0)
= 0.00

Entropy(Recidivist, Drugs) = P(F)*E(Recidivist, F) + P(T)*E(Recidivist, T)

= .8333*(0.970951) + .1667*(0.000)
= .809125

InfoGain(Rec,Drugs) = E(Rec) – E(Rec,Drugs) = 1.00 – 0.809125 = 0.190875

The decision tree splits based on the highest information gain, 0.459, which is “Young” (aka Age < 30).



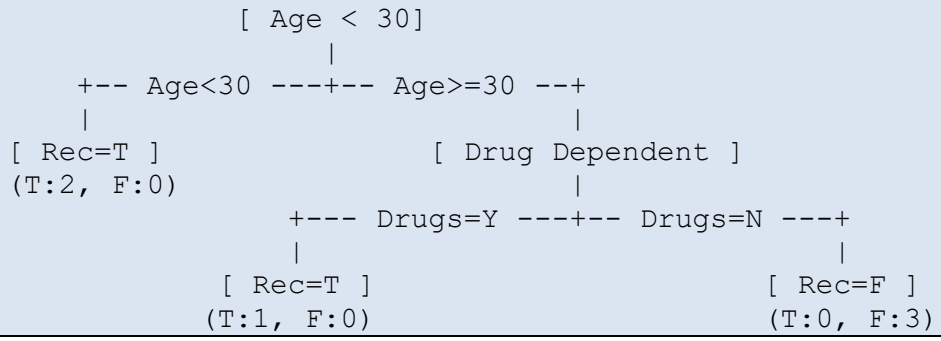
Second Split

The remaining table looks like the following and, by inspection, Drugs (aka the Drug Dependent flag) is a perfect replication of Recidivisty (while Behavior isn’t), so I know with 100% certainty that the remaining node will split on Drugs. I.e. Drugs == T → Recidivist := T and Drugs == F → Recidivist := F.

Behavior	Drugs	Recidivist
F	F	F
T	F	F
T	T	T
T	F	F

The decision tree splits based on the highest information gain, 1.0, which is “Drug Dependent (Drugs)”. ***This also completes the decision tree and the algorithm itself.***

The completed tree is:



Question 3 (More Classification Measures)

I will write on two classification measures: **Matthews Correlation Coefficient (MCC)** and the **False Acceptance Rate (FAR)** measure – the latter of which is used in Biometrics. Biometrics is special in that they cover so-called "two instance" problems which are slightly different than the plain vanilla classification problem.

3.1 Matthews Correlation Coeffecient (MCC)

The Matthews Correlation Coefficient (MCC) is used as a measure of quality of boolean targets: that is, where the class variable can take one of two values, say, "Yes" and "No". It has a close relationship to the **Phi Coefficient** and the Chi-squared statistic -- | MCC | = sqrt (χ^2 / n) -- but I will focus on its other formulation that is derivable directly from the 2x2 confusion matrix which is:

$$MCC = [(TP * TN) – (FP * FN)] \; / \; \text{sqrt} [(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)]$$

MCC ranges from -1.0 to +1.0 while, obviously, |MCC| would range from 0.0 to +1.0 only. The range of MCC from -1.0 to +1.0 is where it gets its correlation moniker. More on this shortly (see properties). A value of 0.00 implies that the predictions are no better than simply guessing randomly.

It is defined such that if any of the products in its denominator are zero, then the denominator is arbitrarily set to 1 and the MCC overall can be set to 0.0 (or simply treated as Div/0 – either way should produce a wary look over of your prediction machine).

Benefits

Consider first that there is no singular number which can describe a 2x2 confusion matrix. This, after all, is why Accuracy, Precision, Recall, F1, etc exist in the first place: they represent various ratios derived from True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).

The MCC, in contrast to those other metrics and accuracy, has some very useful **benefits**:

1. Unlike accuracy and F1, it is robust even when classes are imbalanced (see properties, below);
2. It's often cited as *the most informative single score* that is derived from the confusion matrix;
3. Unlike the oft-cited F1 which places little value on true negatives (TN), the MCC is useful when true negatives are meaningful to a particular analysis.

To emphasize the usefulness of benefit #1 over F1 and Accuracy, consider a set of things where there is truly a 95% : 5% makeup of something. Guessing 100% in the majority class will lead to 95 TP and 5 FP. Note that the F1 score and Accuracy are gushing with pride, whereas the MCC is acting as the canary in

the coalmine that something might be up. The MCC's Division by Zero should draw your attentio to the absence of a zero prediction.

Looking at the case where our prediction machine lazily predicts the majority class:

	Pr:0	Pr:1				
Ac:0	0	5	TN	FP	MCC:	#DIV/0!
Ac:1	0	95	FN	TP	Accuracy:	0.95
					F1 Score:	0.974

Even looking at a more realistic case where our prediction machine still does a pretty lousy job in the presence of a majority class, compare the MCC relative to the overly-generous Accuracy and F1 score:

	Pr:0	Pr:1				
Ac:0	3	5	TN	FP	MCC:	0.15
Ac:1	17	95	FN	TP	Accuracy:	0.817
					F1 Score:	0.896

Formula Inspection

Let's now turn to it's awkward and terrible looking formula:

$$MCC = [(TP * TN) – (FP * FN)] / \text{sqrt} [(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)]$$

First, be mindful that Accuracy can be written as: Acc. = (TP + TN) / (TP + TN + FP + FN), and so this should take some of the sting out of the MCC formula as there are some mathematical similarities: it almost seems like its cousin in the way that a geometric mean is the cousin to an arithmetic mean.

In fact, as a correlation coefficient, it is the geometric mean of the regression coefficients - and its *dual*.

The MCC's formula demonstrates an even-handed view of the various elements of the confusion matrix, ***not emphasizing or valuing TP's more than TN's.*** Instead, it bases it's focus on whether elements are on the main diagonal or not on the main diagonal of the confusion matrix.

Examination of the MCC's Properties

If all elements of the confusion matrix are on the diagonal, then the MCC returns unity:

	Pr:0	Pr:1				
Ac:0	17	0	TN	FP	MCC:	1.00
Ac:1	0	39	FN	TP		

MCC = [(TP * TN) – (FP * FN)] / sqrt [(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)]

Conversely, if the elements of the confusion matrix are ONLY OFF on the diagonal, then MCC = -1.0:

	Pr:0	Pr:1				
Ac:0	0	10	TN	FP	MCC:	-1.00
Ac:1	31	0	FN	TP		

As you may've expected, if there is a balance between the elements of the confusion matrix, then the MCC correlation coefficient returns 0.00, meaning that the prediction may as well be random with respect to the truth:

	Pr:0	Pr:1				
Ac:0	7	7	TN	FP	MCC:	0.00
Ac:1	7	7	FN	TP		

Playing now: if we tip the scales a bit towards on element off the main diagonal, then it goes negative:

	Pr:0	Pr:1				
Ac:0	7	7	TN	FP	MCC:	-0.03
Ac:1	8	7	FN	TP		

If we the off-diagonal is tipped so that it's just less than the diagonal, then the MCC will lean towards the main diagonal and head towards +1.0 with *a slightly positive value:*

	Pr:0	Pr:1				
Ac:0	7	7	TN	FP	MCC:	0.04
Ac:1	6	7	FN	TP		

More importantly, and what may not seem so intuitive is when there is a LARGE class imbalance then it still returns a zero correlation so long as the ratios of the classes remain proportional with each other:

	Pr:0	Pr:1				
Ac:0	2222	8888	TN	FP	MCC:	0.00
Ac:1	1	4	FN	TP		

And, again here is MCC = 0 when the classes are proportional relative to the predicted classes:

	Pr:0	Pr:1				
Ac:0	333	33	TN	FP	MCC:	0.00
Ac:1	777	77	FN	TP		

Online Sources:

https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

https://www.quora.com/Which-is-the-conceptual-difference-between-Matthews-correlation-coefficient-and-F1-score

3.2 Biometrics Performance Measures (eg FAR, FRR, EER)

Biometrics is the science of body measurements and calculations, most often used in authentication and security systems, such as retinal scanners, fingerprint scan devices, facial recognition cameras, etc. We shall see that their terminologies appear different but are really analogs to the ones seen for classification problems.

However, Biometrics quickly descends into the complexities of Multibiometrics:

- Multi-sensors – different angles taken on the same face by different cameras
- Multi-instance – two separate eyes are used to match a single individual
- Multi-samples – A given finger's fingerprint sample is captured twice or thrice.
- Muti-modal – data from different biometric modalities are used to ID (e.g. voice and iris)

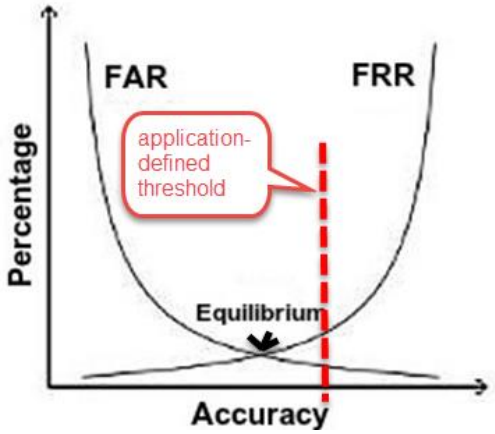
Looking deeper, multiple-instance learning extends the idea of individual sets of labelled instances, there is now a set of bags which each bag containing many instances. The bag has to be marked as either a positive or negative class based on the identity of the instances within the bag.

If all the instances inside the bag are positive or all of them are negative, then the conclusion is trivial. What is done when only SOME are positive while OTHER instances inside a bag are negative? In short, a threshold is usually defined that is then used to tag the bag one way or the other.

Let's return to consider the simple, one-instance case. I intend to just review basic terminology, here.

So, a biometric system can run in two main modes: Identification and Verification. Identification tries to determine a person's identity given their biometric features. In Verification, the person claims an identity beforehand and the biometric features are given to the system to accept the claim or reject the claim. In essence they are usually trying to achieve the same aim: let good people pass and stop bad people from passing.

Not unlike the tradeoff between Precision and Recall, there is a threshold between the False Acceptance Rate (FAR) and False Rejection Rate (FRR). A lazy system could accept everybody (guaranteed zero False Rejection) or reject everybody (guaranteed zero False Acceptance). Somewhere in between there would be 'equilibrium'. A threshold can be defined along these curves as it applies to whether it's better to lean towards False Rejection or False Acceptance.



Source: Adapted from https://www.bayometric.com/false-acceptance-rate-far-false-recognition-rate-frr/

False Acceptance Rate (FAR) is the measure of the likelihood that the biometric system would incorrectly accept an access attempt by an unauthorized user. It's the ratio of count(FalseAcceptances) divided by count(IdentificationAttempts). False Match Rate (FMR) is another word for FAR. While FRR or False Non-Match Rate (FNMR) is complementary to FAR, as discussed above.

There are other performance metrics for Biometrics. To cite one: the Equal Error Rate (EER) - and sometimes called the crossover rate (CER) – these just indicate when a system is operating such that the FAR and FRR are the same. In general a device with a low CER is considered *the most accurate* and running acceptably well.

In conclusion, Biometrics has its own terminologies for performance metrics which have analogs to those we've seen in other classification systems but things can become more complex as different modalities and the problem of multiple-instances is delved into more deeply.

Online Sources:

https://en.wikipedia.org/wiki/Biometrics#Performance

https://www.ncbi.nlm.nih.gov/books/NBK219892/

https://www.bayometric.com/false-acceptance-rate-far-false-recognition-rate-frr/

https://en.wikipedia.org/wiki/Multiple_instance_learning

Question 4 (The Intern: "Which" Measure to Use?)

4.1. Fraud.

If 3% of credit card transactions are truly fraudulent, then the bank would want to identify those correctly (true positives), allowing some but not many false positives as noise that would likely inconvenience and thus annoy good customers. False negatives (fraud gone undetected) would ideally be minimized, but this might be part of the cost of doing business. Fraud can also annoy good customers and potentially be more damaging to them then mere inconveniences.

Let's assume that undetected Fraud – once identified by a customer - are covered by the bank so that customers are quickly made whole and relieved. Then in this case, FN's are acceptable and the focus will be on reducing FP's which can be annoying to good customers day to day purchases: I will assume that a rejected credit card purcahase can be very embarassing for a customer and lead them to leave the bank.

Therefore, we want to minimize FP's whereas FN's aren't so significant to catch.

Given my assumptions, PRECISION would be an ideal metric to track since it punishes a high ratio of FP's to TP's. Compare Illustration 4.1.1 and 4.1.2. Conversely, a HORRIBLE metric to use would be Accuracy, due to the large class imbalance. See Illustration 4.1.3.

Illustration 4.***1.1*** *- High Precision: Many fraudsters can get away but we annoy a small %'ge of customers relative to the fraudsters caught:*

	Pr:0	Pr:1				
Ac:0	97000	1	TN	FP	MCC:	7.63%
Ac:1	2980	19	FN	TP	Accuracy:	97.02%
					F1 Score:	1.26%
					Precision:	95.00%
					Recall:	0.63%

***Illustration** 4.**1.2** - Low Precision: We've caught hundreds of fraudsters but at the expense of hundreds of inconvenienced customers.*

	Pr:0	Pr:1				
Ac:0	97000	500	TN	FP	MCC:	30.58%
Ac:1	2000	500	FN	TP	Accuracy:	97.50%
					F1 Score:	28.57%
					Precision:	50.00%
					Recall:	20.00%

Illustration 4.**1.3** – Due to the large class imbalance, Accuracy would allow a lazy machine to hum away, and is thus a horrible measure to use here:

	Pr:0	Pr:1				
Ac:0	97000	0	TN	FP	MCC:	1.80%
Ac:1	2999	1	FN	TP	Accuracy:	97.00%
					F1 Score:	0.07%
					Precision:	100.00%
					Recall:	0.03%

4.2. Cancer Scans

Let's assume that if you have cancer, that you do want to know about it so that you can begin to do something about it. And while a false diagnosis of cancer (FP) can be unnerving and stressful, any diagnosis of cancer would likely be followed by a second opinion which ideally produce the desireable True Negative (TN) report.

Imagine having cancer and being told that bubbling, black mole on your neck is probably just an ingrown hair when it is actually melanoma. This would be a False Negative and it's highly undesirable, here.

The desired metric would be RECALL (with the effects of a low recall shown in illustration 4.2.1). A HORRIBLE metric would be Precision (see illustration 4.2.2 for an exaggerated case).

Illustration 4.2.1 : The ratio of FN's to TP's here is too high, leading to a relatively low Recall (that is, relative to some desired recall threshold set at, say, 98% or 99%)

	Pr:0	Pr:1					
Ac:0	1000	1	TN	FP	MCC:	89.47%	
Ac:1	6	30	FN	TP	Accuracy:	99.32%	
					F1 Score:	89.55%	
					Precision:	96.77%	
					Recall:	83.33%	<-- Want 98%+

Illustration 4.2.2: Precision would be horrible to use since it seeks only to have correct TP predictions without any concern given towards public safety when cancer is not detected in large numbers of people.

	Pr:0	Pr:1					
Ac:0	0	1	TN	FP	MCC:	-17.70%	
Ac:1	1000	30	FN	TP	Accuracy:	2.91%	
					F1 Score:	5.66%	
					Precision:	96.77%	<-- tone deaf.
					Recall:	2.91%	<-- Want 98%+

4.3. Spam Email Filter

Spam email is an annoyance but it's not the end nof the world when one sneaks by. Far more annoying is not getting an email from an external email address because it ended up in the oft-forgotten Junk folder, at least to me. I don't want ANY good emails going to my junk folder and I want to spank the IT admin with a performance metric that assures that.

For me, Precision is the metric of choice and set with a very high threshold, essentially 100% or 99.99%.

A horrible metric would be Accuracy

Illustration 4.3.1: I want IT Admin to be spanked the moment that Precision drops below 99.99%

	Pr:0	Pr:1					
Ac:0	100	1	TN	FP	MCC:	62.05%	
Ac:1	20	20	FN	TP	Accuracy:	85.11%	
					F1 Score:	65.57%	
					Precision:	95.24%	<-- I want 100%
					Recall:	50.00%	<-- I don't care.

Illustration 4.3.2: The True Negative Rate is completely tone deaf to my screaming to IT about why I'm missing 10 important emails.

	Pr:0	Pr:1					
Ac:0	1000	10	TN	FP	MCC:	96.20%	
Ac:1	8	300	FN	TP	Accuracy:	98.63%	
					F1 Score:	97.09%	
					Precision:	96.77%	
					Recall (aka TPR, Sensitivity):	97.40%	
					Specificity (aka TNR):	0.99%	<-- tone deaf.

4.4. Sports Analytics

When a sports analytics company is betting on "who will win the match" it's usually for high stakes and not just for a friendly gamble about who is going to buy the afternoon donuts.

If we predict them to lose, we want them to lose. If we predict them to win, we want them to win. And that's it. We don't want to hear about how they won when we said they'd lose or vice versa. So Accuracy is key. We don't want to veer off that diagonal of the confusion matrix.

That said, better than accuracy would be the MCC, or Matthews Correlation Coeffecient (see my response to assignment question 3). It provides a measure of the ratio between on-diagonal and off-diagonal items. It's harder to achieve closer to 100% than accuracy and in that respect it's more sensitive than accuracy.

Illustration 4.4.1 Both MCC and Accuracy are good indicators that we're staying on the main diagonal.

	Pr:0	Pr:1					
Ac:0	30	3	TN	FP	MCC:	79.27%	<-- better metric
Ac:1	5	40	FN	TP	Accuracy:	89.74%	<-- good metric
					F1 Score:	90.91%	
					Precision:	93.02%	
					Recall (aka TPR):	88.89%	
					Specificity (aka TNR):	9.09%	

A *horrible* metric would be Precision since it's not paying to the other side of the main diagonal.

Illustration 4.4.2: Precision would be horrible to use unless you're trying to hide your trading losses.

	Pr:0	Pr:1					
Ac:0	1	0	TN	FP	MCC:	3.04%	
Ac:1	55	3	FN	TP	Accuracy:	6.78%	
					F1 Score:	9.84%	
					Precision:	100.00%	<-- hoo boy.
					Recall (aka TPR):	5.17%	
					Specificity (aka TNR):	0.00%	

4.5. Twitter Chatter and Emergency Situations

Emergency resources (ambulance, police) can't spend all day charging around town showing up at false alarms. At the same time, every potential emergency could be a life-threatening one. So this looks like a case for F1 Score, where both FP's and FN's are minimized, more or less equally (it could even be adjusted to weight more towards one or the other but still keeping both low).

A horrible metric would be True Negative Rate, as most chatter on Twitter is bullcrap about kitten videos and daily opinions about Donald Trump -- not related to life-threatening injuries. TNR would lose the FP's and FN's in the deafening clamor of online stupidity and ignorance.

Illustration 4.5.1: Avoiding FPs and FN's could be done with an F1 score with a high threshold (95%+).

	Pr:0	Pr:1				
Ac:0	1000	5	TN	FP	MCC:	81.20%
Ac:1	4	20	FN	TP	Accuracy:	99.13%
					F1 Score:	81.63%
					Precision:	80.00%
					Recall (aka TPR):	83.33%
					Specificity (aka TNR):	0.50%

4.6. Predicting that my MMAI-863 Exam Grade will be Posted

I want my exam grade to be posted and the worse thing for me is checking – thinking that it's gotta be posted by now – and being wrong, again. Every time I check, I want it to be there and I'm very dismayed and upset when it's not. On days I don't think it's there I don't check and there's no problem but I count it nontheless towards the total times I've even thought about it.

I've checked 8 times of the 12 times and was disappointed each time. My True Negative Rate (TNR) is so low: it's 33%. Yeesh. Ideally, I would've either thought to myself "yknow, it's too early" and not even checked or "oh, I should check" and it'd be there.

A horrible metric to use would be Accuracy as I'm not going for the record about guessing how many times it's not there. I could get a metric of 100% by not checking all September and October and then also never looking again, and retire with a perfect but yet-unsatisfying 100% accuracy.

Illustration: 4.6.1: It's like Valentines Day in Elementary School All Over Again.

	Pr:0	Pr:1				
Ac:0	4	8	TN	FP	MCC:	0.02%
Ac:1	0	0	FN	TP	Accuracy:	33.33%
					F1 Score:	0.00%
					Precision:	0.00%
					Recall (aka TPR):	100.00%
					Specificity (aka TNR):	33.33% <-- ouch.

Question 5 (Grocery Association Rules)

Recall: the Rule X -> Y holds with support S if S% of transactions contain X and Y. The Rule X -> Y holds with confidence C if C% of transactions that contain X also contain Y.

a) A rule that has high support and high confidence

Eggs -> Bacon.

The Grocer (Uncle Steve) could infer that eggs and bacon are specific for making breakfast – either standard eggs and bacon or perhaps a quiche, depending on the clientele. (In comparison, "Eggs -> Flour" might be more suggestive of a baker).

Going with the assumption that such high support / common itemset is used for breakfast eaters, the grocery may consider putting nearby to the area with refrigerated foods a display or live presentation with a picture of a 'fancy' breakfast spread made up of **higher-margin** foods -- gourmet breads, fancy jams, fresh squeezed orange juice, prepared fruit salads, etc. to sway some minds that they could add a little pizzazz to their breakfast.

b) A rule that has reasonably high support but low confidence

Cookie Dough -> Ranch Dressing

Hoo boy. This is the calling card of the depressed loner who on Friday night is eating raw cookie dough and ordering pizza and then spooging on loads of rich ranch dressing. Poor fella. Sad, but common.

Well, good thing they come to Steve's Grocery where we will happily *take advantage* of your fragile emotional state to jam higher-end services at you. In this regard, Steve's well-trained check out clerks will slip a coupon to try our premium subscription Home Delivery service. Now, you don't have to shamefully stand in line with your raw cookie dough and ranch cradled in your arms while you wait to silently check out. Now, for the low-low price of $50/month we'll bring your essentials to your door, ring the doorbell, and run off before you open it. No fuss, no muss, so you can get back to the couch.

c) A rule that has low support and low confidence.

Baking Soda -> Vinegar

No, it's not for making volcanos. This type of customer is **eco-conscious** and likes using every day household products to make a variety of homemade cleaning products: vinegar to clean blinds, remove scale from the coffee machine's carafe, use on no-wax floors, hair rinse, and keeping cut flowers fresher longer. Baking soda for teeth brushing, toaster oven fire retardent, deodorant, treating heartburn, and relieving itchy skin. Oh, we know all about you. You probably ride your bicycle everywhere, too, and rub that into people's faces.

Well, we at Uncle Steve's Grocery will send you coupons by mail (or e-coupons) targeted just for you with a whole host of green products that we know you'll want to buy. We've even marked them up for you but then offer bundled prices if you get two or more items on your next visit to our store.

d) A rule that has low support and high confidence

Refried Beans -> Tortilla Chips

You can't hide it, you are a loud and proud **football fan!** We know that you're having people over and making your fabulous **7-layer dip.** You're also social, you have parties (nobody can eat seven layer dip on their own), and you like party recipes. But more than that, we want to let you know that we sell premium beer can wrappers with Steve's Grocery on it with football and other sports themes on them, because you're fun and we're fun. Find them in Aisle 6 next to our homemade, Uncle Steve™ branded beer pong sets (just enough branding to convince you to pay nearly double for what is really just, y'know, plain old ping pong balls and plastic cups). Thank you for shopping at Uncle Steve's and "Goooo Patriots, Yay!"

Question 6 (Wine Club)

We can use similarities between the (n-1) choices to help a person make an informed prediction – or "recommendation" - on the n'th choice of a product.

Using a concept from vector math, we can calculate the angle between two vectors. We will form our vectors from the five (5) wine ratings that we have for all people and do angular distance (CD) computations between Steve and each of the other people in the club.

As a reminder, I provide both the Cosine Similarity and Cosine Distance formulas here to discuss later:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

where A_i and B_i are components of vector A and B respectively.

$D_C(A, B) = 1 - S_C(A, B)$, where D_C is the cosine distance and S_C is the cosine similarity.

Online Source: https://en.wikipedia.org/wiki/Cosine_similarity

The computations (provided below from Excel) show that the cosine distances, CD, for Yuri (CD = 0.02064; Zin rating: 7) and Brigid (CD: 0.02680; Zin score: 5) are very close to each other, nearly the same. So we have some choices, there, which I will discuss presently.

While we could argue that Yuri is the closest and thus Steve should adopt his score of 7, I would prefer the more nuanced approach of taking both into account and basically taking the average, which would predict Steve as giving Zin a rating of "6." …In fact, I did neither….

…*To be farther still*[1], I did a linear regression against all the Zin ratings as measured against the independent variates represented by the CD. I did this partly because I noticed that people who were less like Steve had even lower ratings so it seemed to fit a linear model pretty well. The linear regression results are graphed and shown below. The y-intercept (keep in mind Steve is a cosine distance of 0 from himself and is thus on this Zin graph as the y-intercept) is at 5.96 which I then round to a clean "6".

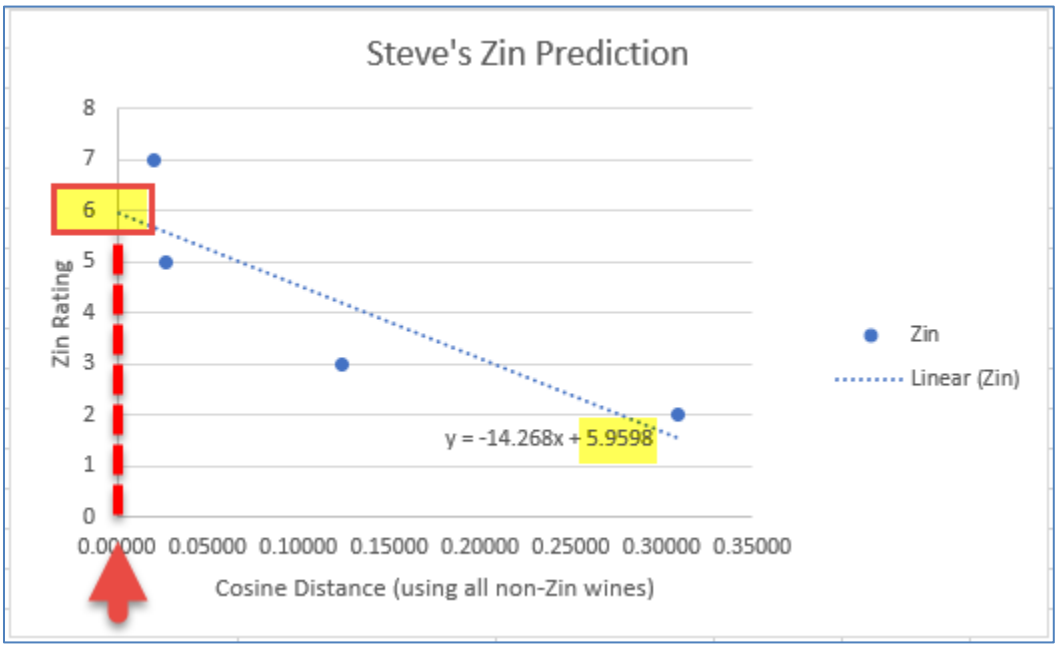
My Final Conclusion: Steve is predicted to rate Zin at a "6".

[1] Actually, I was *initially* thinking a form of weighted average but went with the linear regression with this particular toy dataset. This regression solution wouldn't extend well to other datasets. Consider, for eg, if Qurat's Zin score had been 7 and not 2: then, that would've tilted the regression line such that it would've reduced Steve's Zin prediction. Giving Qurat as much weight as people closer to Steve would've revealed the lack of validity if that had been the case. A weighted average based on some sort of harmonic mean-weight or inverse weighting of the cosine distance was what I'd try if the regression hadn't yielded such a *pretty* and even-handed *looking* result.

Snapshot: Excel calculations for Cosine Distance to Steve.

	Name	P.Noir	Chard	Merlot	Cab	P.Gris	L2 Metric			Zin Prediction
3	Steve	7	6	4	3	4	11.22		"0.00"	????
4								Cosine Similarity (CS) to Steve*	CosineDistance** to Steve = 1 - CS	Zin
5	Yuri	6	7	4	5	4	11.92	0.97936	0.02064	7
6	Gary	3	3	1	1	5	6.71	0.87650	0.12350	3
7	Qurat	2	1	3	7	4	8.89	0.69159	0.30841	2
8	Brigid	6	7	2	3	3	10.34	0.97320	0.02680	5
9										
10	***Cosine Similarity Calculation (for Yuri-Steve):**									
11	`($D$3*$D5 + $E$3*$E5 + $F$3*$F5 + $G$3*$G5 + $H$3*$H5)/($I$3*$I5)`									
12										
13	***Cosine Distance Formula (for Yuri)**									
14	`1-J5`									

Snapshot: Linear Regression to use the ZinRating vs CD to Predict Steve's Zin Rating



Snapshot: Not a meme, but…

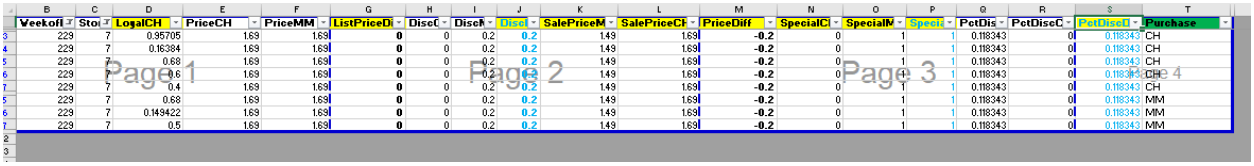
Question 7 (Orange Juice, Classifiers)

The Jupyter notebook and rendered PDF are available as artifacts. I just thought I'd put some of the discussion here where it's easier to read.

EDA, Preprocessing, & Data Engineering:

Rather than use, for example, the raw price of Cirtus Hill (CH) and Minute Maid (MM) I prefered to use the **differences** to explain leans towards one or the other. I did use the raw SalePrices just in case consumers were motivated by the net price and not just the difference in price. In general, I wasn't trying to maximize revenue or items sold (which would've been a different problem) so this is why I just focused on deltas, for the most part. I tossed out most of the nominal price features on that basis.

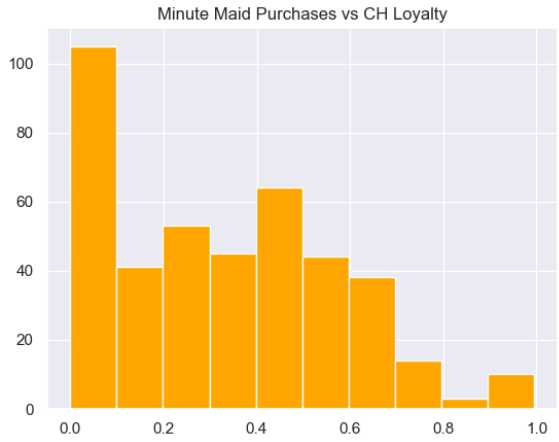
Snapshot: My Excel tab I used to do data engineering (yellow columns were ones selected).

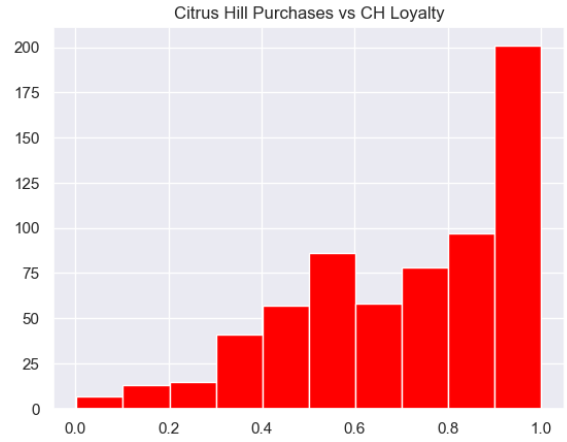


	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	Weekoff	Stor	LogalCH	PriceCH	PriceMM	ListPriceDi	DiscC	DiscM	Disc	SalePriceM	SalePriceCH	PriceDiff	SpecialCl	SpecialM	Speci	PctDis	PctDiscC	PctDiscD	Purchase
3	229	7	0.95705	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	CH
4	229	7	0.16384	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	CH
5	229	7	0.68	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	CH
6	229	7	0.6	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	CH
7	229	7	0.4	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	CH
5	229	7	0.68	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	MM
6	229	7	0.149422	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	MM
7	229	7	0.5	1.69	1.69	0	0	0.2	0.2	1.49	1.69	-0.2	0	1	1	0.118343	0	0.118343	MM

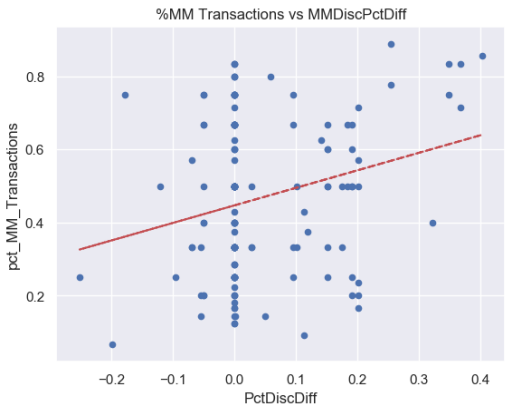
I did want to look at loyalty patterns to see how loyal people were when their prefered OJ was more expensive than the alternative. The **difference in shapes** of the two historgrams were of interest to me and I'd want to know more about that if I had more business context.

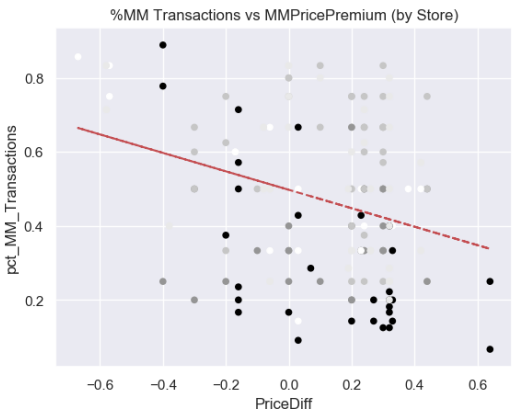
Snapshots: MM and CH Purchases as compared with the CH Loyalty metric.





I also looked at (whether existed and) what the general relationship was between the price difference and the ratio of units sold. Unsurprisingly, the cheaper something was, the more it would be sold. And the higher the price premium relative to the alternative's price then the less something sold. I also wanted to see if there were inter-store differences (greyscale dots) but I couldn't see any that stood out.





Splitting the Data

I used a simple 80%-20% rule to split the data up. I considered using stratified sampling but then hesitated when I had to choose what to stratify based upon – LoyaltyCH, the store, the week, the target. LoyalCH felt best but then it came down to the exact interpretation of the LoyalCH – whether that was known before the time of selection or just at the time of purchase and whether it was for a single customer or an aggregation of customers. I didn't feel great about making assumptions so I just did no stratified sampling at all as a result. Also, I thought the data was regular and didn't have weird clusters of behavior. But that was all the thought that lead me to that conclusion.

I also considered K-fold cross-validation but you had told us to "split the data into training and test sets" so I assumed you explicitly wanted that to be done, removing cross-validation from the options.

Performance Metric

I chose F1 score although plain vanilla accuracy and MCC was also considered for this dataset. I went with F1 score figuring that I wanted to accurately predict the Purchase without being too far off the main diagonal of the Confusion Matrix: I didn't want to overguess towards MM nor CH. I felt F1 covered the bases and was known to the Python functions I was using (MCC wasn't).

Model Tuning

I tuned 6 classifier models; 4 non-ensemble, 2 ensemble. The hyperparameter tuning choices I made were left in as commented out code. I'd try a setting, run, check the performance metric, tinker, rerun. I hadn't really relished this part of the exercise as I had spent a lot of time during data prep and EDA and was getting mentally tired; I just did it enough to witness some very modest changes to the results, typically no more than a few percentage points. For each, I left the confusion matrix and performance metrics (3) for each for reference.

Performance Results Comparison (F1 Score) & Best Model

Results	Classifier	F1 Score	Tiebreaker1	Tiebreaker2
	Decision Tree	F1 Score = 0.82		
	SVM	F1 Score = 0.72		
	K Nearest Neigbors	F1 Score = 0.77		
	Logistical Regression	F1 Score = 0.83		
Best -->	XGBoost	F1 Score = 0.84	Accuracy = 0.84	Kappa = 0.68
A Close Second -->	Random Forest	F1 Score = 0.84	Accuracy = 0.84	Kappa = 0.67

I think the logistical regression and decision tree performed very well, actually. I tend to like non-ensemble methods if they perform similar to ensemble methods as they are more readily interpretable. But, by the numbers, XGBoost came out ahead by a nose over Random Forest, with an F1 of 84%.

Good Enough to Deploy?

Aw, Hell, naw. 'Nothing like writing a ton of code in one evening and sending it onto Production, un-QA'd. I mean, I wasn't sure what the business objective was. To predict purchase, sure, but why so: as an inventory management / reordering exercise? For vendor kickbacks and sales bonus incentives? A model should be measured against KPIs and here there were none. Further, I felt I wanted to know more about LoyalCH and how it was measured; I also wanted to know about number of units sold per transaction before I could do a finer, more engaged modeling job.