MATH 390.4 / 650.2 Spring 2018 Homework #4t

Chaim Eisenbach

Monday 7th May, 2018

Problem 1

These are questions about Silver's book, chapters ... For all parts in this question, answer using notation from class (i.e. $t, f, g, h^*, \delta, \epsilon, e, t, z_1, \ldots, z_t, \mathbb{D}, \mathcal{H}, \mathcal{A}, \mathcal{X}, \mathcal{Y}, X, y, n, p, x_{\cdot 1}, \ldots, x_{\cdot p}$, $x_{1\cdot}, \ldots, x_{n\cdot}$, etc. and also we now have $f_{pr}, h^*_{pr}, g_{pr}, p_{th}$, etc from probabilistic classification as well as different types of validation schemes).

(a) [easy] What algorithm that we studied in class is PECOTA most similar to?

Enhanced Nearest Neighbors.

(b) [easy] Is baseball performance as a function of age a linear model? Discuss.

It's not linear. It's described as an aging curve. Athletes continue to get better until their late 20's then their skills start to deteriorate. (unless you're Ichiro or Bartolo Colon...)

(c) [harder] How can baseball scouts do better than a prediction system like PECOTA?

With experience baseball scouts can see the potential of minor league players even if they are putting up poor stats. Scouts use a hybrid approach. They can go to the field and "see" and "measure" for themselves. They won't be fooled by false positives.

(d) [harder] Why hasn't anyone (at the time of the writing of Silver's book) taken advantage of Pitch f/x data to predict future success?

Scouts are already able to assess pitchers without it. While they didn't have concrete quantitative data they had very good qualitative data.

(e) [difficult] Chapter 4 is all about predicting weather. Broadly speaking, what is the problem with weather predictions? Make sure you use the framework and notation from class. This is not an easy question and we will discuss in class. Do your best.

Even small changes in the x's will drastically change $f(x)$ so it's hard to get a good g. And if the data changes our g will be useless.

(f) [easy] Why does the weatherman lie about the chance of rain? And where should you go if you want honest forecasts?

They are biased, deliberately so, for economic incentive. Presentation over accuracy. Also, they don't want people to get upset when there is rain or something like that. A more accurate source is the National Weather Service.

(g) [difficult] Chapter 5 is all about predicting earthquakes. Broadly speaking, what is the problem with earthquake predictions? It is *not* the same as the problem of predicting weather. Read page 162 a few times. Make sure you use the framework and notation from class.

We cant find a good enough $\mathbb{D}$ to produce a model and the $\mathbb{D}$ we are working with is very noisy and borderline useless. Also, we don't even know what kind of f we are looking for because of our lack of understanding so we can try to get functions that return accurate probabilities but since we don't know what f is supposed to look like it's very difficult to predict.

(h) [easy] Silver has quite a whimsical explanation of overfitting on page 163 but it is really educational! What is the nonsense predictor in the model he describes?

The color of the locks (and the corresponding combinations).

(i) [easy] John von Neumann was credited with saying that "with four parameters I can fit an elephant and with five I can make him wiggle his trunk". What did he mean by that and what is the message to you, the budding data scientist?

If a complex model fits a data set well you should not be impressed. With enough parameters you can fit any data set.

(j) [difficult] Chapter 6 is all about predicting unemployment, an index of macroeconomic performance of a country. Broadly speaking, what is the problem with unemployment predictions? It is *not* the same as the problem of predicting weather or earthquakes. Make sure you use the framework and notation from class.

Because there is no longer a clear connection with economic growth and job growth it's difficult to come up with a working $\mathcal{A}$ i.e. the connection between our x's and our y's has become so complicated we don't know how to connect $\mathcal{H}$ to $\mathbb{D}$ or what even is $\mathcal{H}$.

(k) [E.C.] Many times in this chapter Silver says something on the order of "you need to have theories about how things function in order to make good predictions." Do you agree? Discuss.

sometimes its helpful to be educated or at least have some semblance of understanding so you can put together a useful set of functions. however, sometimes ignorance is better and you can iteratively come up with a solution, while an educated man can spend a long time barking up the wrong tree because of some preconceived notion he has about how something is supposed to work.

Problem 2

This question is about validation for the supervised learning problem with one fixed $\mathbb{D}$.

(a) [easy] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a simple validation and include the final step which is shipping the final g.

1 - $g_i = \mathcal{A}(\mathcal{H}_j, \mathbb{D}_{train})$
2 - Compare $OOSE_j = error(y_{select}, g_j(X_{select})$
3 - Repeat steps 1 & 2 for all $j = (1...n)$ models.
4 - $j^\star = argmin\{oose_1, ..., oose_m\}$
$oose_{j^\star} = error(y_{test}, g_i(X_{select}))$
6 - do 1-4 on $\mathbb{D}$ to produce g.

(b) [easy] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a K-fold cross validation and include the final step which is shipping the final g.

1 - $g_k = \mathcal{A}(\mathcal{H}_j, \mathbb{D}_{train})$
2 - save $\hat{y}_k = g_k(X_{test}, k)$
3 - Repeat 1 & 2 for each fold
4 - Concatenate vertically $\hat{y}_{cv} = \begin{bmatrix} \hat{y}_1 \\ . \\ . \\ . \\ \hat{y}_k \end{bmatrix}$
5 - compute $oose = error(\vec{y}, \vec{\hat{y}}_{cv}$ Randomize order do the whole procedure many times and take the average.

(c) [harder] For one fixed $\mathcal{H}$ and $\mathcal{A}$ (i.e. one model), write below the steps to do a bootstrap validation and include the final step which is shipping the final g.

Create our $\mathbb{D}_{inbag}$(train) by sampling observations in $\mathbb{D}$ with replacement. then we have a g_{train} on $\mathbb{D}_{inbag}$ and run g_{inbag} on $\mathbb{D}_{oob}$ to validate. Compute oos metrics in $\mathbb{D}_{oob}$ with $g_{inbaf}(x_{oob})\&y(oob)$. Run g_{final} on $\mathbb{D}$.

(d) [harder] For one fixed $\mathcal{H}_1, \ldots \mathcal{H}_M$ and $\mathcal{A}$ (i.e. M different models), write below the steps to do a simple validation and include the final step which is shipping the final g.

1 = $g_j = \mathcal{A}(\mathcal{H}_j, \mathbb{D}_{train})$
2 - Compute $oose_j = error(y_{select}, g_j(x_{select}))$
3 - repeat steps 1 & 2 for all j = 1....M.
4 - $j^\star = argmin\{oose_1, ..., oose_m\}$
5 - $oose_{j^\star} = error(y_{test}, g_{j^\star}(x_test))$.
6 - build final g using $\mathbb{D}$

(e) [difficult] For one fixed $\mathcal{H}_1, \ldots \mathcal{H}_M$ and $\mathcal{A}$ (i.e. M different models), write below the steps to do a K-fold cross validation and include the final step which is shipping the final g. This is not an easy problem! There are a lot of steps and a lot to keep track of...

1 - $g_{jk_ik_0} = \mathcal{A}(\mathcal{H}_j, \mathbb{D}_{train}, k_ik_0)$
2 - save $\hat{y}_{jk_ik_0} = g_{jk_ik_0}(\mathbb{D}, k_i, k_0)$
3 - Repeat 1 & 2 for all models j $\in$ 1...m
4- repeat 1 & 2 for all inner folds $k_i \in \{1...m\}$
5 - Concatenate vertically $\hat{y}_{j,k_0} = \begin{bmatrix} \hat{y}_{j_1,k_0} \\ . \\ . \\ . \\ \hat{y}j_m, k_0 \end{bmatrix}$
6 - select best medol $j^\star_{k_0} = argmin\{oose_{j,k_0} \ ... \ oose_{m,k_0}\}$
b 7 -Repeat steps 1-6 for all $k_0 \in \{1....M\}$
8 - set $\hat{\vec{y}} = \begin{bmatrix} \hat{y}_{j^\star_1,1} \\ . \\ . \\ . \\ \hat{y}j^\star_m, m \end{bmatrix}$
9 - estimate $oose = error(\hat{\vec{y}}, \vec{\hat{y}})$
10 - Repeat 1 - 6 to build final model g without $\mathbb{D}_{test}$ (only inner CV loop)

Problem 3

This question is about ridge regression — an alternative to OLS.

(a) [harder] Imagine we are in the "Luis situation" where we have $\boldsymbol{X}$ with dimension $n \times (p+1)$ but $p+1 > n$ and we still want to do OLS. Why would the OLS solution we found previously break down in this case?

If X is not full rank then $X^\top X$ is not invertible and there is no unique solution for b.

(b) [harder] We will embark now to provide a solution for this case. The solution will also give nice results for other situations besides the Luis situation as well. First, assume λ is a positive constant and demonstrate that the expression $\lambda \left|\left|\boldsymbol{w}\right|\right|^2 = \boldsymbol{w}^\top(\lambda \boldsymbol{I})\boldsymbol{w}$ i.e. it can be expressed as a quadratic form where $\lambda \boldsymbol{I}$ is the determining matrix. We will call this term $\lambda \left|\left|\boldsymbol{w}\right|\right|^2$ the "ridge penalty".

$\lambda \left|\left|\boldsymbol{w}\right|\right|^2 = \lambda w^\top w = w^\top \lambda I w$

(c) [easy] Write the $\mathcal{H}$ for OLS below where there parameter is the $\boldsymbol{w}$ vector. $\boldsymbol{w} \in$?
$\mathcal{H} = \{X * w, w \in \mathbb{R}^{p+1}\}$

(d) [easy] Write the error objective function that OLS minimizes using vectors, then expand the terms similar to the previous homework assignment.
$(y - Xw)^\top (y - Xw)$ Expanded below.

(e) [easy] Now add the ridge penalty $\lambda \left|\left|\boldsymbol{w}\right|\right|^2$ to the expanded form you just found and write it below. We will term this two-part error function the "ridge objective".

$(y - Xw)^\top (y - Xw) + \lambda w^\top w \Rightarrow y^\top y - X^\top w^\top y - y^\top Xw + w^\top X^\top Xw + \lambda w^\top w \Rightarrow$
$y^\top y - 2w^\top X^\top y + w^\top (X^\top X + \lambda I) w$

(f) [easy] Note that the ridge objective looks a bit like the hinge loss we spoke about when we were learning about support vector machines. There are two pieces of this error function in counterbalance. When this is minimized, describe conceptually what is going on.

The first part is to minimize how far off our predictions were. The second part prevents overfitting by making it harder for w to get large

(g) [harder] Now, the ridge penalty term as a quadratic form can be combined with the last term in the least squares error from OLS. Do this, then use the rules of vector derivatives we learned to take $d/d\boldsymbol{w}$ and write the answer below.

$-2X^\top y + 2(X^\top X + \lambda I)w = 0$

(h) [easy] Now set that derivative equal to zero. What matrix needs to be invertible to solve?

$X^\top X + \lambda I$ Needs to be invertible

(i) [difficult] There's a theorem that says *positive definite* matrices are invertible. A matrix is said to be positive definite if every quadratic form is positive for all vectors i.e. if $\forall \boldsymbol{z} \neq \boldsymbol{0} \;\; \boldsymbol{z}^\top A \boldsymbol{z} > 0$ then A is positive definite. Prove this matrix from the previous question is positive definite.

$z^\top (X^\top X + \lambda I) z = z^\top (X^\top X) z + z^\top (\lambda I) z = ||Xz||^2 \lambda \sum z_i^2 > 0$ Since both parts have to be greater than zero.

(j) [easy] Now that it's positive definite (and thus invertible), solve for the $\boldsymbol{w}$ that is the argmin of the ridge objective, call it $\boldsymbol{b}_{ridge}$. Note that this is called the "ridge estimator" and computing it is called "ridge regression" and it was invented by Hoerl and Kennard in 1970.

$b_{ridge}(X^\top X + \lambda I)^{-1} X^\top y$

(k) [easy] Did we just figure out a way out of Luis's situation? Explain.

We control for overfitting through ridge regression even when p+1 > n we have a model.

(l) [harder] It turns out in the Luis situation, many of the values of the entries of $\boldsymbol{b}_{ridge}$ are close to 0. Why should that be? Can you explain now conceptually how ridge regression works?

We can't give much importance to each feature since we don't have enough data to be confident in the weight of our predictors Ridge regression prevents overfitting by limiting the weights so our model doesn't get out of control. By minimizing $||w||$ we keep our weights under control.

(m) [easy] Find $\hat{\boldsymbol{y}}$ as a function of $\boldsymbol{y}$ using $\boldsymbol{b}_{ridge}$. Is $\hat{\boldsymbol{y}}$ an orthogonal projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$?

$\hat{y} = Xb_{ridge}$ And it is still and orthogonal projection.

(n) [E.C.] Show that this $\hat{\boldsymbol{y}}$ is an orthogonal projection of $\boldsymbol{y}$ onto the column space of some matrix $\boldsymbol{X}_{ridge}$ (which is not $\boldsymbol{X}$!) and explain how to construct $\boldsymbol{X}_{ridge}$ on a separate page.

(o) [easy] Is the $\mathcal{H}$ for OLS the same as the $\mathcal{H}$ for ridge regression? Yes/no.
Is the $\mathcal{A}$ for OLS the same as the $\mathcal{A}$ for ridge regression? Yes/no.

Yes. No

(p) [harder] What is a good way to pick the value of λ, the hyperparameter of the $\mathcal{A}$ = ridge?

Make a lot of λ and make a ridge regression model for each and using model selection find the best λ

(q) [easy] In classification via $\mathcal{A}$ = support vector machines with hinge loss, how should we pick the value of λ? Hint: same as previous question!

Make a lot of λ and make a SVM model for each and using model selection find the best λ

(r) [E.C.] Besides the Luis situation, in what other situations will ridge regression save the day?

When we want polynomial terms we can limit how large the high degree polynomials are allowed to become and not look crazy.

(s) [difficult] The ridge penalty is beautiful because you were able to take the derivative and get an analytical solution. Consider the following algorithm:

$$\boldsymbol{b}_{lasso} = \underset{\boldsymbol{w} \in \mathbb{R}^{p+1}}{\arg\min} \left\{ (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda ||\boldsymbol{w}||^1 \right\}$$

This penalty is called the "lasso penalty" and it is different from the ridge penalty in that it is not the norm of $\boldsymbol{w}$ squared but just the norm of $\boldsymbol{w}$. It turns out this algorithm (even though it has no closed form analytic solution and must be solved numerically a la the SVM) is very useful! In "lasso regression" the values of $\boldsymbol{b}_{lasso}$ are not shrunk *towards* 0 they are harshly punished *directly to* 0! How do you think lasso regression would be useful in data science? Feel free to look at the Internet and write a few

sentences below.

It might weed out the "useless predictors" This way we can look at the more important predictors which have a stronger relationship to y

(t) [easy] Is the $\mathcal{H}$ for OLS the same as the $\mathcal{H}$ for lasso regression? Yes/no.
Is the $\mathcal{A}$ for OLS the same as the $\mathcal{A}$ for lasso regression? Yes/no.
Yes. Yes.

Problem 4

These are questions about non-parametric regression.

(a) [easy] In problem 1, we talked about schemes to validate algorithms which tried M different prespecified models. Where did these models come from?

We are smart and can estimate the types of models that could potentially be useful. Stepwise model construction.

(b) [harder] What is the weakness in using M pre-specified models?

You are stuck with only those models. Like a director who requires his/her actors to stick to the script instead of allowing them to improvise. Your movie will be as scripted yet you may not have gotten the actors best performances. Misspecification error.

(c) [difficult] Explain the steps clearly in forward stepwise linear regression.

Starting with no variables in the model. Testing the addition of each variable using a chosen model fitting our requirements. Adding the variable whose inclusion gives significant improvement to the fit. And repeating the process until none improves the model in ant significant way.

(d) [difficult] Explain the steps clearly in *backwards* stepwise linear regression.

Starting with all candidate variables, testing the deletion of each variable using a chosen model, deleting any variable whose loss is insignificant and repeating until all such variables are gone.

(e) [harder] What is the weakness(es) in this stepwise procedure?

You still need to specify the predictor set and the model is still linear.

(f) [easy] Define "non-parametric regression". What problem(s) does it solve? What are its goals? Discuss.

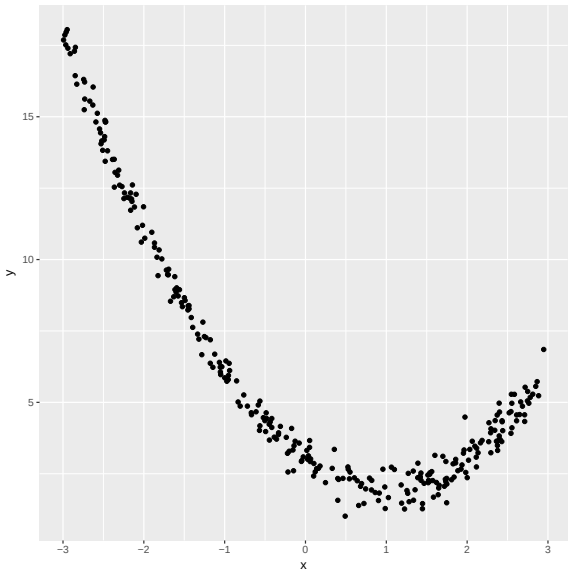
Non-Parametric Regression: The model will not be of the fixed form but constructed

according to the data. $\mathcal{H}$ will adjust flexibly allowing for optimal expressiveness in the $\mathbb{D}$

(g) [harder] Provide the steps for the regression tree (the one algorithm we discussed in class) below.

1- Beginning with all the training data.
2- For every possible split at the current node divide data into x_l, $\vec{y_l}$ and $x_r, \vec{y_r}$. Calculate $SSE_l = \sum(y_l - \bar{y_l})^2$ for r and l.
3 - Find the split with the total lowest SSE which is sum of r and l SSE.
4 - Create the split. Split data into two nodes.
5 - Repeat steps 2-4 until "STOP".
STOP: node has $\leq N_0$ data points inside. default N_0 is $= 5$

(h) [easy] Consider the following data

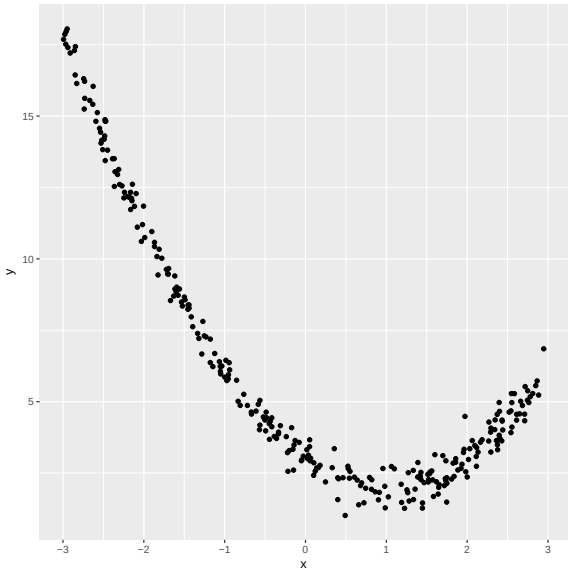


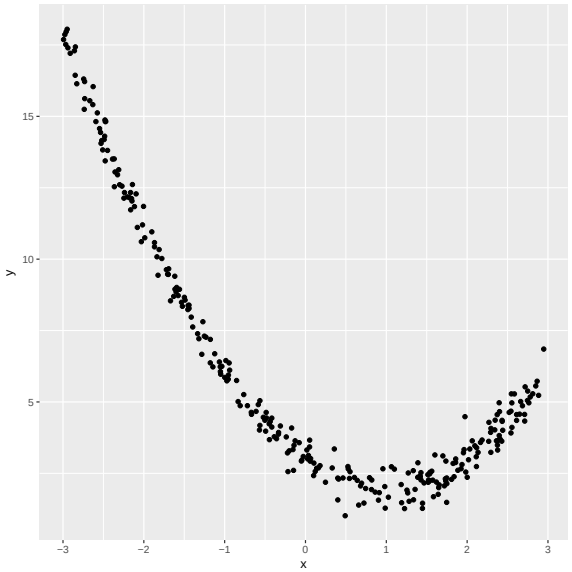
Create a tree with maximum depth 1 (i.e one split at the root node) and plot g above.

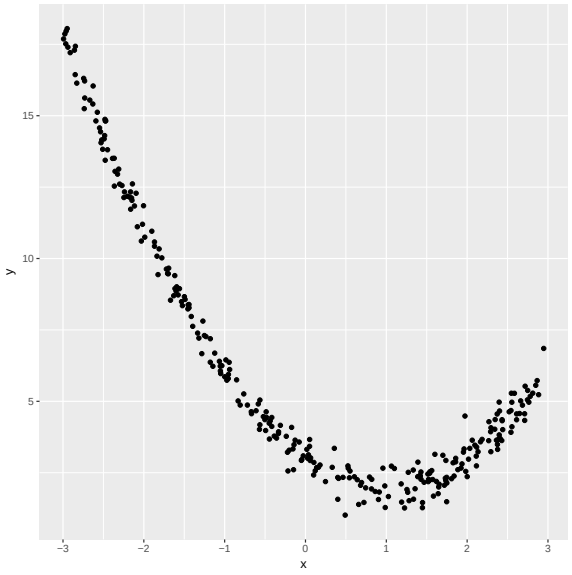
(i) [easy] Now add a second split to the tree and plot g below.

(j) [easy] Now add a third split to the tree and plot g below.

(k) [easy] Now add a fourth split to the tree and plot g below.

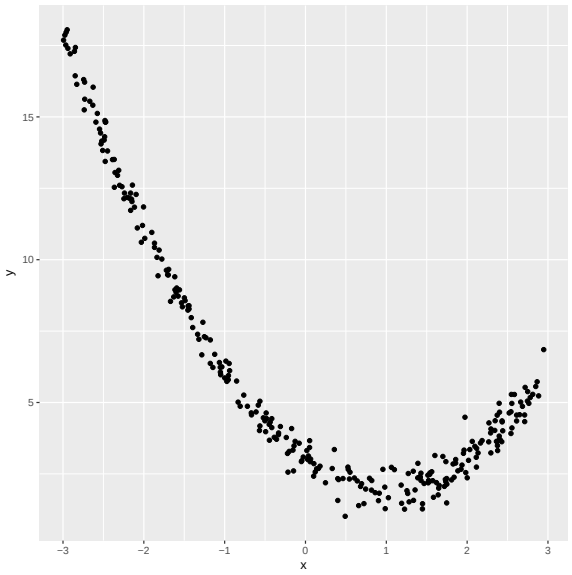

15
10
5
y
−3
−2
−1
0
1
2
3
x


15
10
5
y
−3
−2
−1
0
1
2
3
x



(l) [easy] Draw a tree diagram of g below indicating which nodes are the root, inner nodes and leaves. Indicate split rules and leaf values clearly.

(m) [easy] Plot g below for the mature tree with the default $N_0 =$ `nodesize` hyperparameter.



(n) [easy] If $N_0 = 1$, what would likely go wrong?
Overfitting.

(o) [easy] How should you pick the $N_0 =$ `nodesize` hyperparameter in practice?
Use model selection or the default if you're lazy.

Problem 5

These are questions about classification trees.

(a) [easy] How are classification trees different than regression trees?

Classification trees work just like regression trees, only they try to predict a discrete category (i.e. class) rather than a numerical value.

(b) [harder] What are the steps in the classification tree algorithm?

1 -Begin with all training data.
2 - For every possible split, calculate the "Gini Impurity" metrics. $Gini_l = \sum_{l=1} k\hat{p}_l(1 - \hat{p}_l, Gini_r = \sum_{l=1} k\hat{p}_r(1 - \hat{p}_r)$ where $\hat{p}_l = \frac{\#y_i incategory}{n_j \#innode}$
3 - Find the split with the lowest weighted average Gini impurity metrics: $Gini_{avg} = \frac{n_l gini_l = n_r gini_r}{n_l + n_r}$
4 - Create the split and split the data in the node to the left right daughter nodes.
5 - Repeat 2-4 until "STOP"
6 - For all leaf nodes assign $\bar{y} = mode[\vec{y_0}]$ where $\vec{y_0}$ are the avg. of the y_i's in the last node.

Problem 6

These are questions about measuring performance of a classifier.

(a) [easy] What is a confusion table?

A table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Consider the following in-sample confusion table where "> 50K" is the positive class:

```
          y_hats_train
y_train  <=50K  >50K
  <=50K   3475   262
  >50K     471   792
```

(b) [easy] Calculate the following: n (sample size) = 5,000

FP (false positives) = 262

TP (true positives) = 792

FN (false negatives) = 471

TN (true negatives) = 3475

$\#P$ (number positive) = 1263

$\#N$ (number negative) = 3737

$\#PP$ (number predicted positive) = 1054

$\#PN$ (number predicted negative) = 3946

$\#P/n$ (prevalence / marginal rate / base rate) = 0.25

$(FP + FN)/n$ (misclassification error) = 0.147

$(TP + TN)/n$ (accuracy) = 0.85

$TP/\#PP$ (precision) = 0.75

$TP/\#P$ (recall, sensitivity, true positive rate, TPR) = 0.627

$2/(\text{recall}^{-1} + \text{precision}^{-1})$ (F1 score) = 0.68

$FP/\#PP$ (false discovery rate, FDR) = 0.249

$FP/\#N$ (false positive rate, FPR) = 0.07

$FN/\#PN$ (false omission rate, FOR) = 0.12

$FN/\#P$ (false negative rate, FNR) = 0.37

(c) [easy] Why is FPR also called the "false alarm rate"?

Because you tell someone they have cancer when they don't

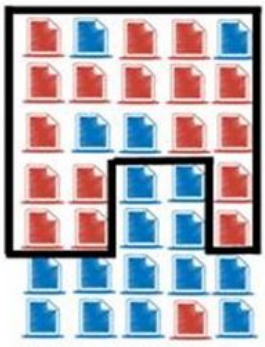
(d) [easy] Why is FNR also called the "miss rate"?

Because you tell someone who has cancer that they don't.

(e) [easy] Below let the red icons be the positive class and the blue icons be the negative class.

The icons included inside the black border are those that have $\hat{y} = 1$. Compute both precision and recall.

Precision = $\frac{17}{21}$. Recall $\frac{17}{18}$



(f) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at FPR vs. FNR. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

In the cancer scenario it would be better to have high FPR and low FNR because untreated cancer is worse than going for more tests.

(g) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at FDR vs. FOR. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

FDR gives us the percentage of how our PP is correct. FOR tells us the percentage of PN that are true.

(h) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look at precision vs. recall. Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

In the cancer we would rather have high recall and not such great precision.

(i) [harder] There is always a tradeoff of FP vs FN. However, in some situations, you will look only at an overall metric such as accuracy (or $F1$). Describe such a classification scenario. It does not have to be this income amount classification problem, it can be any problem you can think of.

We want to balance the recall and precision. It tells us how effective our cancer test is.