Andrew Li-Yang Liu
UNI: ALL2209
Date due: Oct 8, 2021
COMS 4771 Machine Learning

Homework 1

Problem 1:

1.1.a: A and B are known (A=a, B=b).

No chance of having $a + b < 0$ since the variables are exponentially distributed. If $0 \leq a + b < 7$:

$$P[Y = 1|A = a, B = b] = P[C < 7 - a - b]$$

$$= \int_0^{7-a-b} e^{-x}dx = -e^{-x}|_0^{7-a-b} = 1 - e^{a+b-7}$$

If $a + b \geq 7$, then clearly $P[Y = 1|A = a, B = b] = 0$. So:

$$P[Y = 1|A = a, B = b] = \begin{cases} 1 - e^{a+b-7}, & \text{if a + b} < 7 \\ 0, & \text{if a + b} \geq 7 \end{cases}$$

$$P[Y = 0|A = a, B = b] = \begin{cases} e^{a+b-7}, & \text{if a + b} < 7 \\ 1, & \text{if a + b} \geq 7 \end{cases}$$

So, if $a + b \geq 7$, Bayes classifier classifies 0. If $a + b < 7$, then we classify 1 if $1 - e^{a+b-7} \geq e^{a+b-7}$ and 0 if $1 - e^{a+b-7} < e^{a+b-7}$. Solving this inequality gives:

$$f^*(A, B, C) = \text{argmax}_y P[Y = y|A = a, B = b]$$
$$= \begin{cases} 1, & \text{if a + b} < 7 - \ln(2) \\ 0, & \text{if a + b} \geq 7 - \ln(2) \end{cases}$$

Now we want to calculate Bayes error:

$$P^*[e|A, B] = P[Y = 1, f^* = 0|A, B] + P[Y = 0, f^* = 1|A, B]$$
$$= P[Y = 1|A, B]P[f^* = 0|A, B] + P[Y = 0|A, B]P[f^* = 1|A, B]$$

Note the following:

$$P[f^* = 1|A, B] = \begin{cases} 1, if\ a + b < 7 - \ln(2) \\ 0, if a + b \geq 7 - \ln(2) \end{cases}$$

$$P[f^* = 0|A, B] = \begin{cases} 0, if\ a + b < 7 - \ln(2) \\ 1, if a + b \geq 7 - \ln(2) \end{cases}$$

So:

$$P[Y=1, f^*=0|A,B] = P[C<7-a-b]P[f^*=0]$$
$$= \begin{cases} 1-e^{a+b-7}, & \text{if } 7-\ln(2) \le a+b \le 7 \\ 0, & \text{if a + b} < 7-\ln(2) \\ 0, & \text{if a + b} > 7 \end{cases}$$

$$P[Y=0, f^*=1|A,B] = \begin{cases} e^{a+b-7}, & \text{if } a+b < 7-\ln(2) \\ 0, & \text{if a + b} \ge 7-\ln(2) \end{cases}$$

$$P^*[e|A,B] = \begin{cases} 0, & \text{if a + b} > 7 \\ 1-e^{a+b-7}, & \text{if } 7-\ln(2) \le a+b \le 7 \\ e^{a+b-7}, & \text{if } a+b < 7-\ln(2) \end{cases}$$

1.1.b: A is known, B and C unknown.

Here, I am letting f be the pdf. If $a<7$:

$$P[Y=1|A=a] = P[B+C<7-a]$$
$$= \int_0^{7-a} P[C<7-a-b]f_B(b)db$$
$$= \int_0^{7-a} [1-e^{a+b-7}]e^{-b}db$$
$$= \int_0^{7-a} [e^{-b}-e^{a-7}]db$$
$$= [-e^{-b}-be^{a-7}]|_0^{7-a}$$
$$= 1-e^{a-7}+(a-7)e^{a-7}$$
$$= 1+(a-8)e^{a-7}$$

If $a \ge 7$ then clearly $P[Y=1|A=a]=0$.

So:

$$P[Y=1|A=a] = \begin{cases} 1+(a-8)e^{a-7}, & \text{if a} < 7 \\ 0, & \text{if a} \ge 7 \end{cases}$$

Then:

$$P[Y=0|A=a] = \begin{cases} (8-a)e^{a-7}, & \text{if a} < 7 \\ 1, & \text{if a} \ge 7 \end{cases}$$

In the case where a<7, Bayes classifier will classify 1 if $1+(a-8)e^{a-7} > (8-a)e^{a-7}$. Solving this inequality (via graphing calculator) gives:

$$f^*(A,B,C) = \begin{cases} 1, & \text{if a} < 5.322 \\ 0, & \text{if a} \ge 5.322 \end{cases}$$

Note:

$$P[f^* = 1|A,B] = \begin{cases} 1, if\ a < 5.322 \\ 0, if\ a \geq 5.322 \end{cases}$$
$$P[f^* = 0|A,B] = \begin{cases} 0, if\ a < 5.322 \\ 1, if\ a \geq 5.322 \end{cases}$$

So:

$$P[Y = 1, f^* = 0|A,B] = \begin{cases} 1 + (a-8)e^{a-7}, & \text{if } 5.322 \leq \text{a} < 7 \\ 0, & \text{if a} \geq 7 \\ 0, & \text{if a} < 5.322 \end{cases}$$
$$P[Y = 0, f^* = 1|A,B] = \begin{cases} (8-a)e^{a-7}, & \text{if a} < 5.322 \\ 0, & \text{if a} \geq 5.322 \end{cases}$$

So:

$$P^*[e|A,B] = \begin{cases} 1 + (a-8)e^{a-7}, & \text{if } 5.322 \leq \text{a} < 7 \\ 0, & \text{if a} \geq 7 \\ (8-a)e^{a-7}, & \text{if a} < 5.322 \end{cases}$$

1.1.c: A, B, and C are all unknown.

$$\begin{aligned} P[Y = 1] &= P[A + B + C < 7] \\ &= \int_0^7 P[B + C < 7 - a] f_A(a) da \\ &= \int_0^7 [1 + (a-8)e^{a-7}] e^{-a} da \\ &= \int_0^7 [e^{-a} + (a-8)e^{-7}] da \\ &= \left[-e^{-a} + \left(\frac{1}{2}a^2 - 8a\right) e^{-7} \right]\Big|_0^7 \\ &= 1 - e^{-7} + \left(\frac{49}{2} - 56\right) e^{-7} \\ &= 0.97036 \end{aligned}$$

Then:

$$P[Y = 0] = 0.02964$$

So, Bayes classifier f* is:

$$f^*(A,B,C) = \text{argmax}_y P[Y = y] = 1$$

So Bayes error is:

$$P^*[e] = 0.02964$$

1.2:

$$\begin{aligned} P^*[e|A,B] &= P[Y = 1, f^* = 0|A,B] + P[Y = 0, f^* = 1|A,B] \\ &= P[C < 7 - a - b] P[f^* = 0|A,B] + P[C \geq 7 - a - b] P[f^* = 1|A,B] \\ &= P[C < 7 - a - b](1 - P[f^* = 1|A,B]) + P[C \geq 7 - a - b] P[f^* = 1|A,B] \end{aligned}$$

Note that for any combination of A,B, only one of the above terms survive, since $P[f^* = 1|A, B]$ can only take values 0 or 1 (given A and B, the classifier deterministically classifies 0 or 1). Hence, the Bayes error can take only take two values:

$$P^*[e|A, B] = \begin{cases} P[C < 7 - a - b], & \text{if } P[f^* = 1|A, B] = 0 \\ P[C \geq 7 - a - b], & \text{if } P[f^* = 1|A, B] = 1 \end{cases}$$

We can easily make the probability for both cases go to ½ by choosing a distribution for C where 7-a-b is the median of the distribution.

Problem 2:

Notation:

- P[e] = nearest neighbor error rate = 1 – accuracy
- $D_n = (X_n, Y_n)$ = labelled training data
- x_{ni} = ith nearest neighbor (i=1,2,3)
- x_{ni} = label of ith nearest neighbor
- (x_t, y_t) = test data and label
- y* = optimal prediction from Bayes' classifier

As was seen in class:

$$\begin{aligned} &\lim_{n\to\infty} P_{y_t, D_n}[e|x_t] \\ &= \lim_{n\to\infty} \int P_{y_t, Y_n}[e|x_t, X_n] P[X_n|x_t]\, dX_n \\ &= \lim_{n\to\infty} \int P_{y_t, y_n}[e|x_t, x_{ni}] P[x_{ni}|x_t] dx_{ni} \end{aligned}$$

For 3-NN, this becomes:

$$\lim_{n\to\infty} \int [1 - P(y_t = \text{majority}(y_{n1}, y_{n2}, y_{n3})|x_t]] P[x_{ni}|x_t] dx_{ni}$$

$$= \lim_{n\to\infty} \int \left[1 - \sum_y P(y_t = y, \text{majority}(y_{n1}, y_{n2}, y_{n3}) = y|x_t]\right] P[x_{ni}|x_t] dx_{ni}$$

$$= \lim_{n\to\infty} \int \left[1 - \sum_y P[y_t = y|x_t] P[\text{majority}(y_{n1}, y_{n2}, y_{n3}) = y|x_t]\right] P[x_{ni}|x_t] dx_{ni} \qquad (*)$$

Let's find $P[\text{majority}(y_{n1}, y_{n2}, y_{n3}) = y|x_t]$ by considering all possible combinations:

$$P[\text{majority}(y_{n1}, y_{n2}, y_{n3}) = y|x_t]$$

$$= P(y_{n1} = y)P(y_{n2} = y)P(y_{n3} \neq y) + P(y_{n1} = y)P(y_{n2} \neq y)P(y_{n3} = y) \\ + P(y_{n1} \neq y)P(y_{n2} = y)P(y_{n3} = y) + P(y_{n1} = y)P(y_{n2} = y)P(y_{n3} = y)$$

Using the fact that $P(y_{ni} \neq y) = 1 - P(y_{ni} = y)$, and taking the limit as $n \to \infty$, which means $y_{ni} \to y_t$ for all i:

$$\lim_{n\to\infty} P[\text{majority}(y_{n1}, y_{n2}, y_{n3}) = y|x_t] = 3P^2(y_t = y)\big(1 - P(y_t = y)\big) + P^3(y_t = y) \\ = 3P^2(y_t = y) - 2P^3(y_t = y)$$

Now plug this term back into (*), and use the fact that $\lim_{n\to\infty} P[x_{ni}|x_t] = P[x_t|x_t] = 1$. Then the limit/integral reduces to:

$$\lim_{n\to\infty} P_{y_t,D_n}[e|x_t] = 1 - \sum_y [3P^3(y_t = y) - 2P^4(y_t = y)]$$

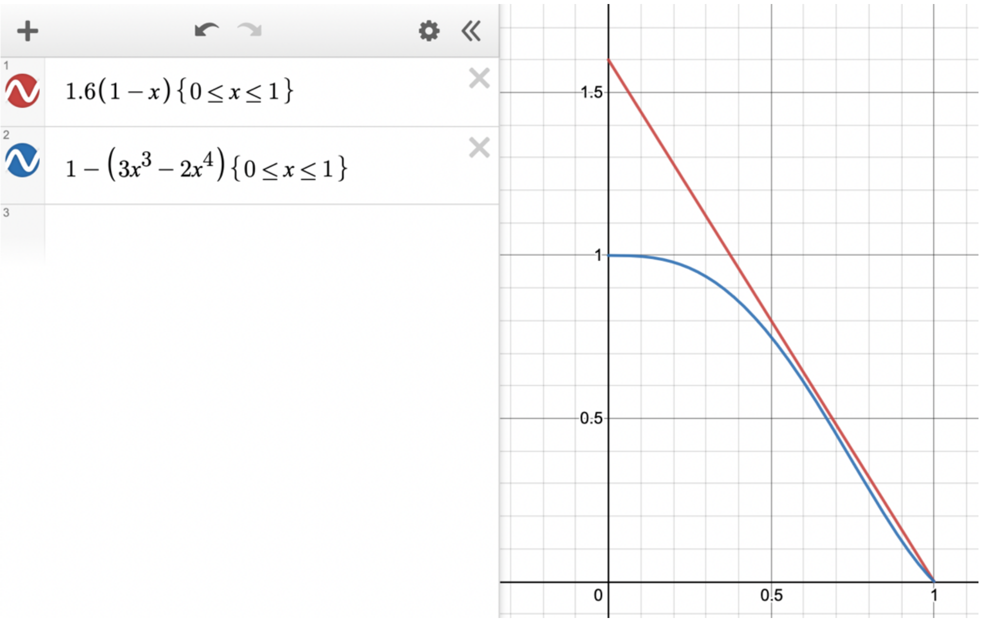
This error is bounded by only taking the Bayes' prediction term from the sum:

$$1 - \sum_y [3P^3(y_t = y) - 2P^4(y_t = y)] \leq 1 - [3P^3(y_t = y^*) - 2P^4(y_t = y^*)]$$

The right hand side of this inequality is in turn bounded by our desired bound:

$$1 - [3P^3(y_t = y^*) - 2P^4(y_t = y^*)] \leq 1.6\big(1 - P(y_t = y^*)\big) = 1.6P^*[e|x_t]$$

We can see that this is indeed an upper bound by plotting the two $1 - [3P^3(y_t = y^*) - 2P^4(y_t = y^*)]$ and $1.6\big(1 - P(y_t = y^*)\big)$ against $P(y_t = y^*)$, where $0 \leq P(y_t = y^*) \leq 1$:



1 $1.6(1-x)\{0 \le x \le 1\}$

2 $1-(3x^3-2x^4)\{0 \le x \le 1\}$

3

(Graph made with Desmos.com)

In other words, the error of 3-NN is at most 1.6 times Bayes error.

Problem 3:

i. Taylor's Remainder Theorem:

For any $a, b \in \mathbb{R}$, $\exists z \in [a, b]$ such that:

$$f(b) = f(a) + f'(a)(b-a) + \frac{1}{2}f''(z)(b-a)^2$$

Assume $\exists L \geq 0$ such that for all $a, b \in \mathbb{R}$: $|f'(a) - f'(b)| \leq L|a-b|$ or $\frac{|f'(a)-f'(b)|}{|a-b|} \leq L$

Then, take the limit as $a \to b$, and the assumption becomes:

$$|f''(z)| \leq L$$

(for all z)

Let $a = \overline{x} := x - \eta f'(x)$, where $\eta > 0$, and $b = x$. Then, based on Taylor's Remainder Theorem:

$$f(x) - f(\overline{x}) = f'(\overline{x})f'(x)\eta + \frac{1}{2}f''(z)[f'(x)]^2\eta^2 \qquad (*)$$

I want to prove that there always exists some $\eta > 0$ such that the RHS is non-negative. **First, we can immediately tell that if $f'(x) = 0$, then the RHS is zero, so $f(x) = f(\overline{x})$.**

Now, note that I can use the regularity assumption to place bounds on $f'(\overline{x})$:

$$|f'(x) - f'(\overline{x})| \leq L|x - \overline{x}| = L|f'(x)|\eta$$

$$-L|f'(x)|\eta \leq f'(x) - f'(\overline{x}) \leq \ L|f'(x)|\eta$$

$$f'(\overline{x}) \geq f'(x) - L|f'(x)|\eta$$

Plug this inequality into (*), and we get:

$$f(x) - f(\overline{x}) \geq \frac{1}{2} f''(z)[f'(x)]^2\eta^2 + [f'(x) - L\eta|f'(x)|]f'(x)\eta$$

$$= \left[\frac{1}{2} f''(z)[f'(x)]^2 - Lf'(x)|f'(x)|\right]\eta^2 + [f'(x)]^2\eta$$

This is a quadratic in η. This quadratic has zeros at:

$$\eta_1 = 0$$

$$\eta_2 = \frac{[f'(x)]^2}{Lf'(x)|f'(x)| - \frac{1}{2} f''(z)[f'(x)]^2}$$

Taking $|f''(z)| \leq L$, or rather $f''(z) \geq -L$, then:

$$\eta_2 \geq \frac{[f'(x)]^2}{L[f'(x)|f'(x)| + \frac{1}{2}[f'(x)]^2]}$$

Now we consider different cases. We already considered the case where $f'(x) = 0$, where we get an equality.

Now, consider $f'(x) < 0$. In this case, the leading coefficient of the quadratic in η is:

$$\frac{1}{2} f''(z)[f'(x)]^2 - Lf'(x)|f'(x) \geq -\frac{1}{2} L[f'(x)]^2 - Lf'(x)|f'(x) = \frac{1}{2}[f'(x)]^2 L > 0$$

In other words, if $f'(x) < 0$, then the leading coefficient of the quadratic (the coefficient of η^2) is positive. This means that we can always choose a sufficiently large η such that $f(x) - f(\overline{x}) = f'(\overline{x})f'(x)\eta + \frac{1}{2} f''(z)[f'(x)]^2\eta^2 > 0$, since when η is large the quadratic term dominates.

Now consider the case where $f'(x) > 0$. In this case, we observe that the second root (η_2) of the quadratic satisfies:

$$\eta_2 \geq \frac{[f'(x)]^2}{L[f'(x)|f'(x)| + \frac{1}{2}[f'(x)]^2\,]} = \frac{[f'(x)]^2}{\frac{3}{2}L[f'(x)]^2} > 0$$

In other words, the quadratic $f(x) - f(\overline{x}) = f'(\overline{x})f'(x)\eta + \frac{1}{2}f''(z)[f'(x)]^2\eta^2$ has a positive root. This means the parabola has to cross the η-axis at a positive root $\eta_2 > 0$, meaning there must be some $\eta > 0$ such that the quadratic is positive, i.e. $f(x) - f(\overline{x}) > 0$.

After considering all cases, we conclude that there exists a $\eta > 0$ such that $f(\overline{x}) \leq f(x)$ for all $x \in \mathbb{R}$, given the regularity assumption and Taylor's remainder theorem.

ii. The gradient descent algorithm:

```
$x^{(0)} = x_0$
While $f'(x^{(t)}) \neq 0$:
        $x^{(t+1)} := x^{(t)} - \eta f'(x^{(t)})$
```

(in practice, the conditional is While $|f'(x^{(t)})| > \delta$, for some small δ)

iii. Using the above algorithm, I found:

$$x^* = 1.074$$
$$f(x^*) = 14.416$$

Problem 4:

Part 4i:

Simply removing the sensitive attribute generally will not work because many other variables may have unforeseen correlations with the sensitive attribute and may thus serve as a proxy for the sensitive attribute. For example, if your objective is to predict whether or not a student graduates, race may be a sensitive attribute. Even if you ignore race, other features such as income level and the neighborhood the student lives in may all serve as proxy features correlated with race. As a result, the demographic parity, equalized odds, and predictive parity may all still be different across race.

Part 4ii:

Start with

$$P_0[\hat{Y} = 1] = P_1[\hat{Y} = 1]$$

Note that

$$P_a[\hat{Y} = 1] = P[\hat{Y} = 1|A = a] = \frac{P[\hat{Y} = 1, A = a]}{P[A = a]}$$

So

$$\frac{P[\hat{Y} = 1, A = 0]}{P[A = 0]} = \frac{P[\hat{Y} = 1, A = 1]}{P[A = 1]}$$

The marginal probability is the sum of the joint probabilities:

$$\begin{aligned}
&P[\hat{Y} = 1] = P[\hat{Y} = 1, A = 0] + P[\hat{Y} = 1, A = 1] \\
&= P[\hat{Y} = 1, A = 0] + \frac{P[A = 1]}{P[A = 0]} P[\hat{Y} = 1, A = 0] \\
&= P[\hat{Y} = 1, A = 0] + \frac{1 - P[A = 0]}{P[A = 0]} P[\hat{Y} = 1, A = 0] \\
&= \frac{P[\hat{Y} = 1, A = 0]}{P[A = 0]} \\
&= P_0[\hat{Y} = 1]
\end{aligned}$$

Similarly:

$$\begin{aligned}
&P[\hat{Y} = 1] = P[\hat{Y} = 1, A = 0] + P[\hat{Y} = 1, A = 1] \\
&= \frac{P[A = 0]}{P[A = 1]} P[\hat{Y} = 1, A = 1] + P[\hat{Y} = 1, A = 1] \\
&= \frac{P[\hat{Y} = 1, A = 1]}{P[A = 1]} \\
&= P_1[\hat{Y} = 1]
\end{aligned}$$

Hence:

$$P_0[\hat{Y} = 1] = P_1[\hat{Y} = 1] \Rightarrow P[\hat{Y} = 1] = P_a[\hat{Y} = 1]$$
$$\forall a \in \{0,1\}$$

Now prove the other way. Start with:

$$P[\hat{Y} = 1] = P_a[\hat{Y} = 1], \forall a \in \{0,1\}$$

Then clearly:

$$P[\hat{Y} = 1] = P_0[\hat{Y} = 1] = P_1[\hat{Y} = 1]$$

So:

$$P[\hat{Y} = 1] = P_a[\hat{Y} = 1], \forall a \in \{0,1\} \Longrightarrow P_0[\hat{Y} = 1] = P_1[\hat{Y} = 1]$$

In conclusion:

$$P_0[\hat{Y} = 1] = P_1[\hat{Y} = 1] \Longleftrightarrow P[\hat{Y} = 1] = P_a[\hat{Y} = 1]$$

$$\forall a \in \{0,1\}$$

Part 4iii:

Let $\boldsymbol{r} \in \mathbb{R}, \boldsymbol{k} \in \mathbb{N}$. Start with:

$$P_1[\hat{Y} = r] = P_2[\hat{Y} = r] = P_3[\hat{Y} = r] = \cdots$$

We see that:

$$P[\hat{Y} = r] = \sum_{k\in\mathbb{N}} P[\hat{Y} = r, A = k]$$

$$= \sum_{k\in\mathbb{N}} P_{a_k}[\hat{Y} = r]P[A = k]$$

But all the $P_k[\hat{Y} = r]$ are equal, so:

$$= P_k[\hat{Y} = r]\sum_{k\in\mathbb{N}} P[A = k]$$

$$= P_k[\hat{Y} = r]$$

So:

$$P_1[\hat{Y} = r] = P_2[\hat{Y} = r] = P_3[\hat{Y} = r] = \cdots \Rightarrow P[\hat{Y} = r] = P_k[\hat{Y} = r]$$

Similarly, if you start with $P[\hat{Y} = r] = P_k[\hat{Y} = r]$, then we easily see:

$$P[\hat{Y} = r] = P_1[\hat{Y} = r] = P_2[\hat{Y} = r] = P_3[\hat{Y} = r] = \cdots$$

So:

$$P_1[\hat{Y} = r] = P_2[\hat{Y} = r] = P_3[\hat{Y} = r] = \cdots \Leftrightarrow P[\hat{Y} = r] = P_k[\hat{Y} = r]$$

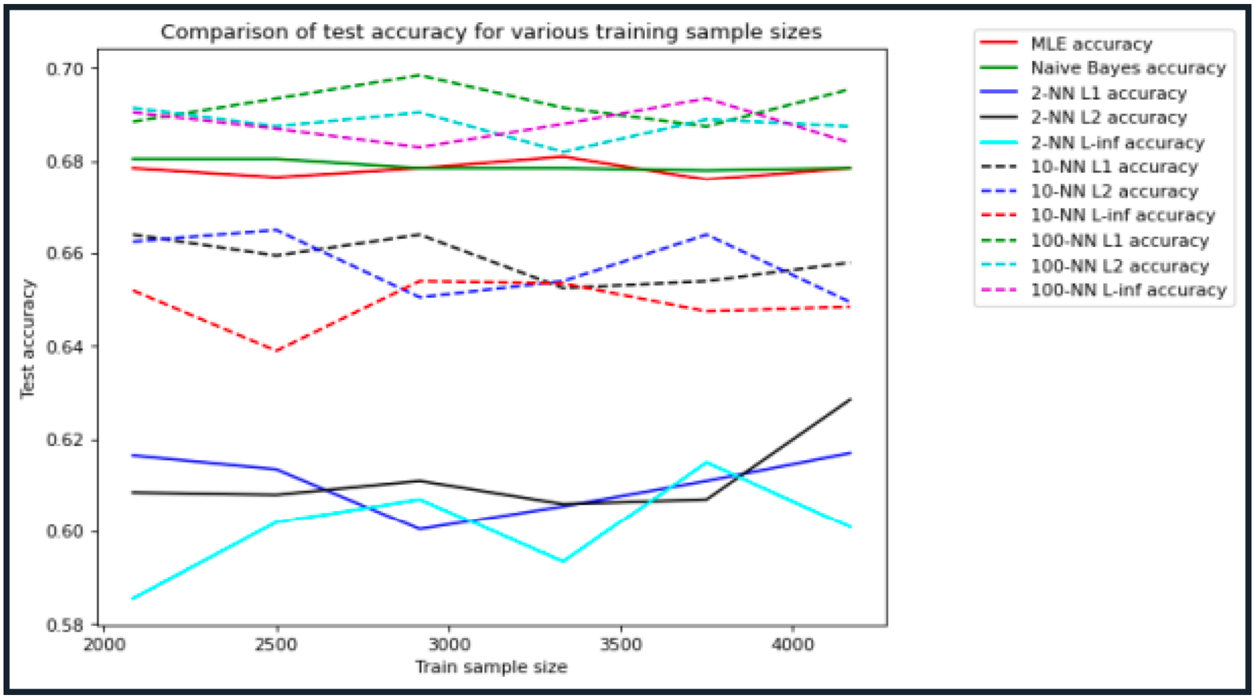
Part 4iv: (see code on separate file)

Part 4v:

In this comparison I compared the test accuracies of the MLE classifier, Naïve Bayes classifier, and the kNN classifier (with k=2,10,100 and p-norm p=1,2, infinity).

I found that all classifiers tend to have rather fairly constant test accuracy with respect to training sample size, which took values between 2083 (50% of original train data) and 4167 (100% of original train data). It seems that, generally, with the exception of at sample_size=3750, the 100-NN L1 classifier performed the best.

It is worth noting though, that the accuracy of the different classifiers seem to be split into three strata. Those at the top strata include all the 100-NN classifiers, as well as the MLE and Naïve Bayes. Within this strata, MLE and Naïve Bayes tend to have very close accuracies, whereas the 100-NN classifiers tend to do slightly better. In the second (middle) strata, we have all the 10-NN classifiers. Finally, at the lowest performance strata we have the 2-NN classifiers.

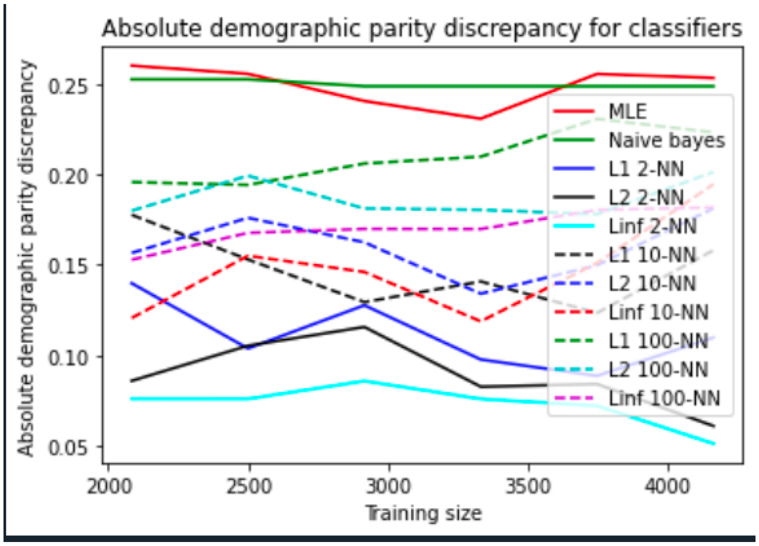


Part 4vi:

For each fairness definition, I simply took the right hand side of the equation, subtracted the other side, and took the absolute value of this discrepancy as a measure for lack of fairness.

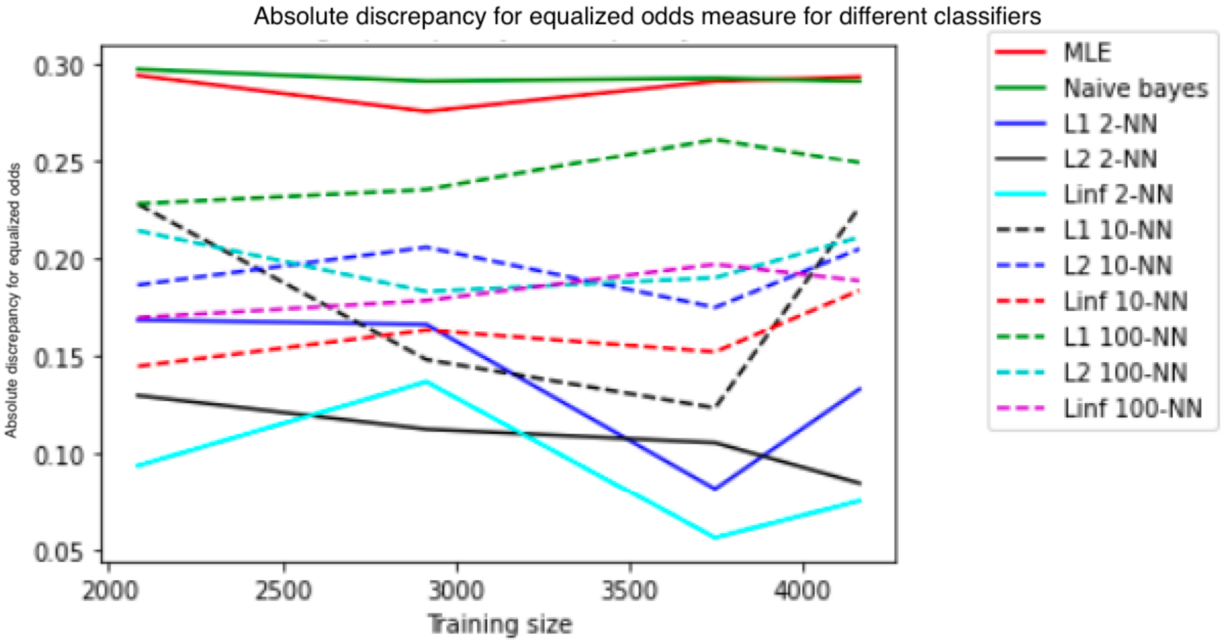
Demographic parity:

For demographic parity, we measure the lack of fairness of each classifier by taking the absolute value of the difference $|P_1[\hat{Y}=1]-P_0[\hat{Y}=1]|$ and compare them as we vary the training size. We find that the infinite-norm 2-NN consistently yields the lowest discrepancy (roughly constant at 0.7), and hence is the most fair. The MLE and Naïve Bayes classifiers did the worst, with a discrepancy of about 0.25.

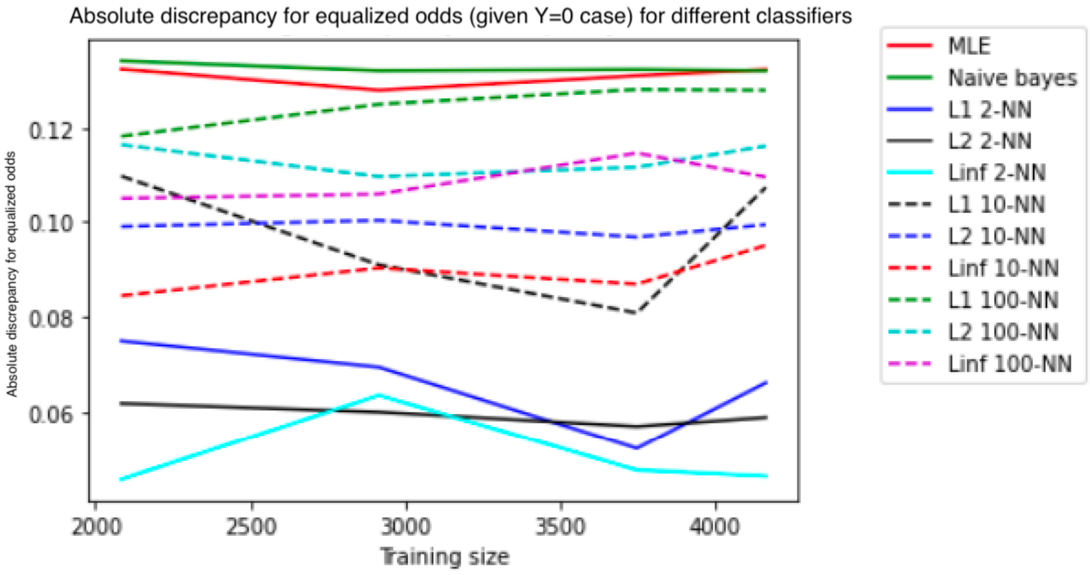


Equalized odds:

For equalized odds, I computed the lack of fairness for each classifier using the absolute value of the difference $|P_0[\hat{Y} = 1|Y = 1] - P_1[\hat{Y} = 1|Y = 1]|$ (due to computational time constraints, we only considered the case where we are given Y=1). In this case, I found that, in general, the infinite-norm 2-NN still tends to score the lowest discrepancy (about around 0.1, hence the highest level of fairness). The MLE and Naïve Bayes classifiers tend to do the worst, with a discrepancy of around 0.30.

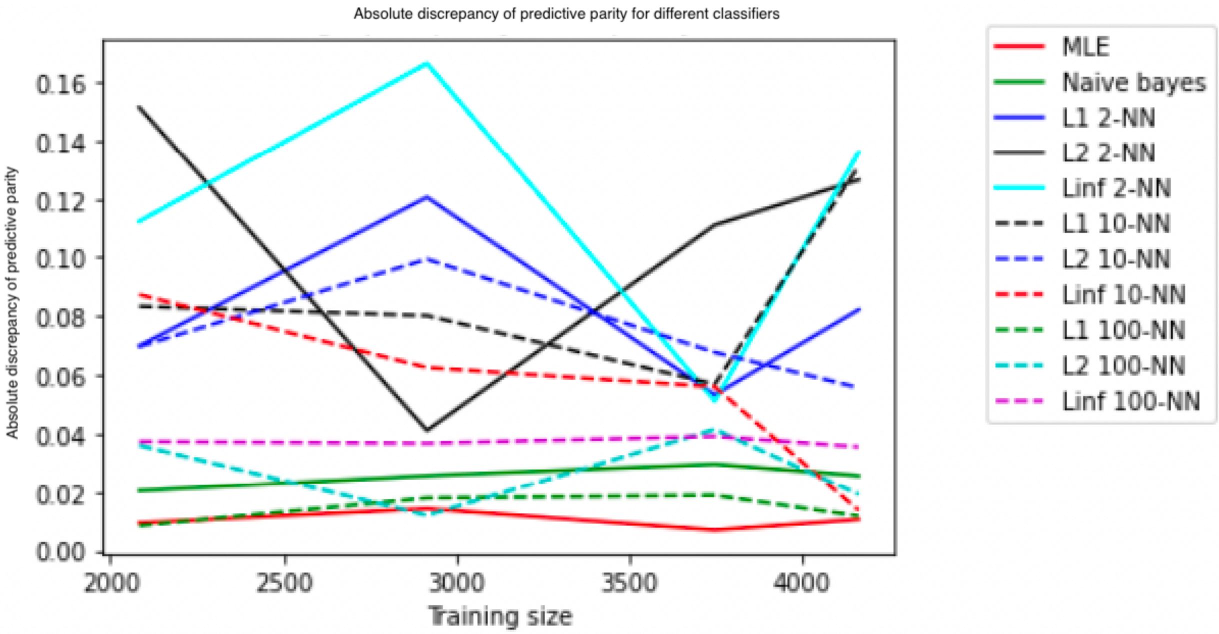


To be sure, we see that the results are similar for the case where we are given Y=0. In this case, we measure the difference $|P_0[\hat{Y} = 1|Y = 0] - P_1[\hat{Y} = 1|Y = 0]|$.

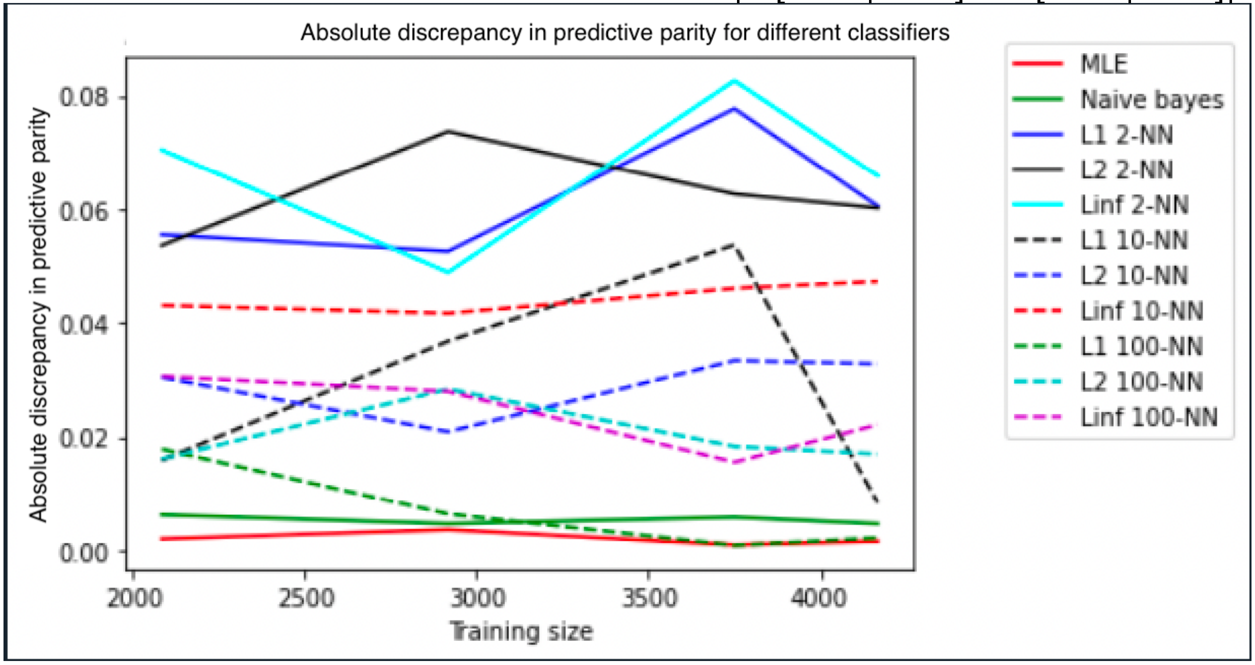


Predictive parity:

In the case of predictive parity, I measured the discrepancy $\left|P_0\left[Y=1\middle|\hat{Y}=1\right]-P_1\left[Y=1\middle|\hat{Y}=1\right]\right|$. It turns out that, here, the fairness rankings seem to be reversed, with the MLE consistently scoring highest on fairness (lowest discrepancy of about 0.01), followed by the L1 100-NN classifier (with discrepancy also around 0.01). The infinite-norm 2-NN seems to do the worst, with its discrepancy oscillating dramatically as training size increases.



This ordering for predictive parity performance seems to be preserved if we measure the discrepancy given the prediction = 0, i.e., if we measure $\left|P_0\left[Y=1\middle|\hat{Y}=0\right]-P_1\left[Y=1\middle|\hat{Y}=0\right]\right|$.



Part 4vii:

Demographic parity may be most reasonable in the case of social welfare programs, where the goal is to benefit underprivileged groups. For example, if the objective of the classification task is to determine whether or not someone should receive emergency funding, the probability of someone qualifying should solely depend on his/her living conditions and not on what race he/she belongs to.

On the other hand, demographic parity fails to account for cases where there are real differences across the sensitive attribute. For example, in Moritz Hardt's article "Approaching fairness in machine learning", he brings up the case of a luxury hotel chain promoting itself to a group of wealthy white people and a group of less affluent black people. The demographic parity condition would force the same percentage of people in both groups to receive the hotel's promotion, even though the less affluent group has no way to engage in a transaction. As such, demographic parity compromises both fairness (in the sense that those qualified to receive the promotion don't receive it), and, by neglecting the differences across the sensitive attribute (in this case, income associated with different races), fails to achieve classification accuracy and profit maximization.

Furthermore, Alexandre Landeau's article "Measuring fairness in machine learning models" states that "[demographic parity] concerns only the final outcome of the model but not on equality of treatment," in the sense that nothing prevents a trained model might be used to select candidates from a majority group, whereas those in a minority group are selected more or less at random, as long as the percentages selected in each group matches.

Sources cited:

Hardt, Moritz. “Approaching Fairness in Machine Learning.” *Moody Rd Blog*, 6 Sept. 2016, http://blog.mrtz.org/2016/09/06/approaching-fairness.html.

Landeau, Alexandre. “Measuring Fairness in Machine Learning Models.” *Medium*, Data from the Trenches, 18 Sept. 2020, https://medium.com/data-from-the-trenches/measuring-fairness-in-machine-learning-models-2be070fab712.

Part 4viii: (not graded)

Problem 5:

Part 5i: (see code on separate file)

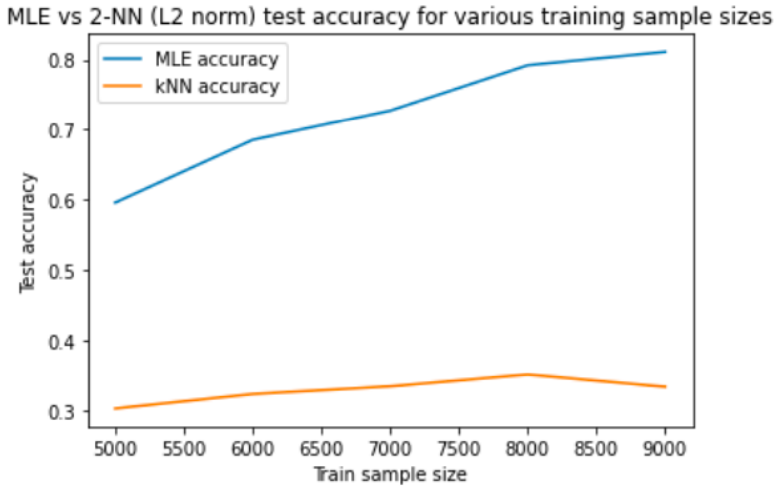
Part 5ii: (see code on separate file)

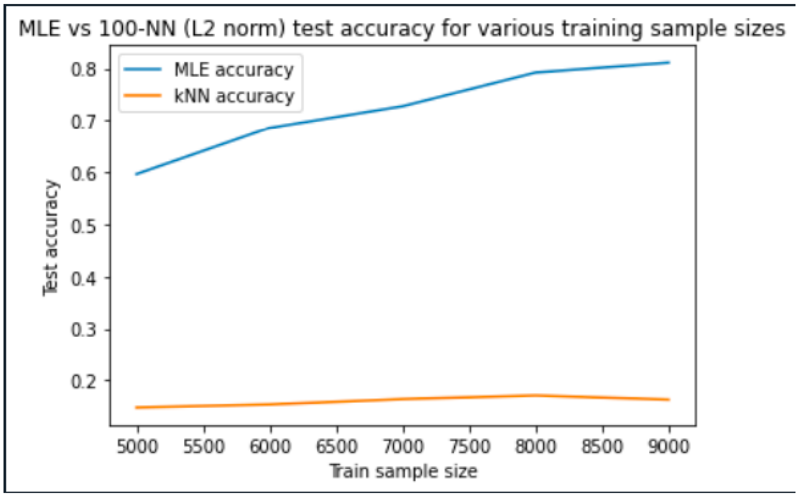
Part 5iii:

Assume we are using L2 norm for the k-NN classifier for this problem. We will only do the comparisons for k=2 and k=100 to save time.

We conclude that the test accuracy of the MLE classifier is consistently higher than that of the kNN classifier. At 5000 training samples, the MLE starts at roughly 60% test accuracy, then as we increase the train sample size to 9000, the test accuracy grows dramatically to over 80%.

The test accuracy of the L2 norm 2-NN classifier, on the other hand starts with accuracy of around 30% and does not increase significantly. The 100-NN does even worse, starting at 15% test accuracy and increasing only up to 16%.

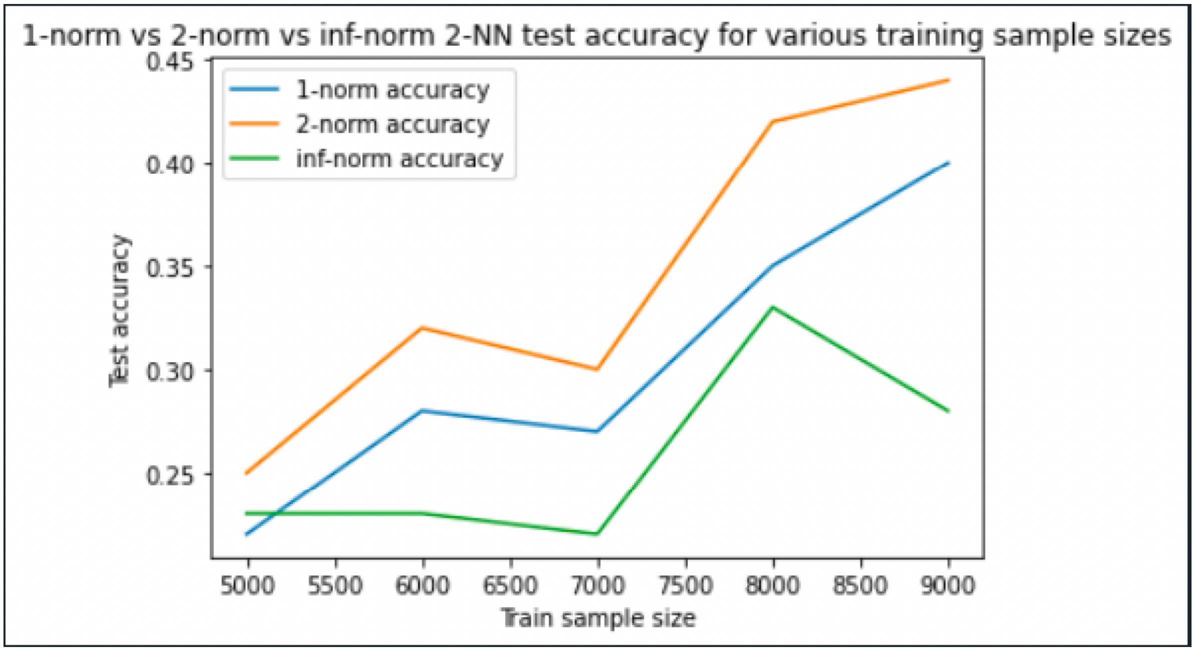




Part 5iv:

To save time, I ran a comparison test on only 100 test samples for each classifier. We also set a constant k=2.

The results show that the 2-norm classifier consistently outperforms the others in terms of test accuracy. The 2-norm accuracy increases from 25% to 44%. The 1-norm comes second (increasing from 22% to 40%), and the infinite-norm one is the worst (increasing from 23% to around 30%, then dropping to around 27%).


1-norm vs 2-norm vs inf-norm 2-NN test accuracy for various training sample sizes
1-norm accuracy
2-norm accuracy
inf-norm accuracy
Test accuracy
0.45
0.40
0.35
0.30
0.25
5000
5500
6000
6500
7000
7500
8000
8500
9000
Train sample size