(1 point for easily legible writing on this cover sheet.)

NetID: _____ (mine is `jgillett` from `jgillett@wisc.edu`)

Last name: _____ First name: _____

Mark your lecture with "X":

_____ TuTh 9:30-10:45

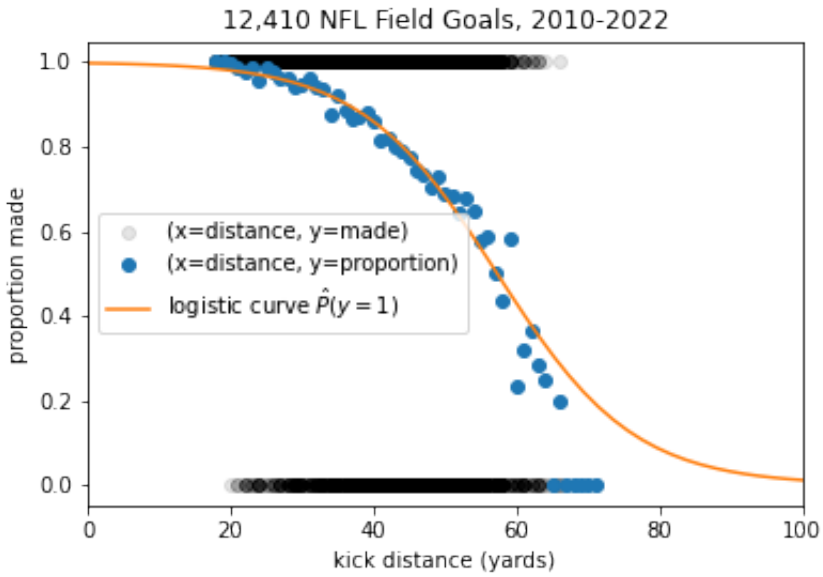
_____ TuTh 11:00-12:15

STAT 451 Midterm Exam Instructions:

1. Please sit in alternating columns.
2. Do not open the exam until I say "go."
3. Put away everything except a pencil or pen, a calculator, and your one-page (two sides) notes sheet.
4. Show your work. Correct answers without at least a minimal version of the work normally required may receive no credit.
5. If a question is ambiguous, resolve the ambiguity in writing. We will consider grading accordingly.
6. The exam ends when I call time. If you continue writing after I call time, you risk a penalty. (The alternative, that you get more time than your peers, is unfair.)
7. If a question is ambiguous, resolve the ambiguity in writing. We will consider grading accordingly. e.g.
   - I think "average" refers to the population mean μ (not the sample mean $\bar{X}$).
   - I think "linear regression" refers to OLS, not ridge or lasso.

Question	Points	Earned
Q0 (cover)	1	
Q1	4	
Q2	5	
Q3	15	
Q4	14	
Q5	10	
Q6	9	
Q7	20	
Q8	10	
Q9	12	
Total	100	

1. From the logistic regression model represented in the figure, estimate the likelihood of an NFL field goal kicker making three field goals in a row, one from 20 yards, one from 40, and one from 60. We may suppose these attempts are independent and make other reasonable simplifiying assumptions.



P(NFL kicker makes the three field goals) = _____

ANSWER:

The likelihood is

$L_{\mathbf{w},b} = \prod_{i=1}^{N} l_{\mathbf{w},b}(\mathbf{x_i}, y_i)$

$= P(y=1|x=20) \cdot P(y=1|x=40) \cdot P(y=1|x=60)$

$\approx 1 \cdot 0.9 \cdot 0.4 = 0.36.$

Any answer in $[0.2, 0.5]$ is ok too.

2. Consider applying gradient descent with step size $\alpha = 0.5$ to find the $\mathbf{x}$ that minimizes the function $f(\mathbf{x}) = f\left(x^{(1)}, x^{(2)}, x^{(3)}\right) = \left(x^{(1)} - 1\right)^2 + \left(x^{(2)} + 2\right)^2 + \left(3 - x^{(3)}\right)^2$ starting from $\mathbf{x}_0 = (1, 1, 1)$. Find the value $\mathbf{x}_1$ after one iteration.

ANSWER:

$\nabla f(\mathbf{x}) = \left(2(x^{(1)} - 1), 2(x^{(2)} + 2), 2(3 - x^{(3)})(-1)\right)$, which is $(0, 6, -4)$ at $\mathbf{x} = (1, 1, 1)$.

Move to $\mathbf{x}_1 = \mathbf{x}_0 - \alpha \nabla f(\mathbf{x}_0) = (1, 1, 1) - 0.5(0, 6, -4) = (1, -2, 3)$.

3. Here are some questions on support vector machines.

(a) Suppose we have a soft-margin SVM for which $\mathbf{w} = (2, 3)$ and $b = -1$. How does the SVM classify $(\mathbf{x} = (2, 1), y = 1)$?

ANSWER:

$\mathbf{wx} + b = (2, 3) \cdot (2, 1) + (-1) = 6 > 0$, so $\hat{y} = 1$.

(b) Suppose we have some training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ in matrices `X` and `y`. We have plotted the data with $y = -1$ examples red and $y = 1$ examples blue. Which line of code gives the best model for predicting new examples?

For each question, write the best answer from among these lines labeled "A" thorugh "H". You may not use an answer more than once.

A: `clf = svm.SVC(kernel='linear', C=1); clf.fit(X, y)`

B: `clf = svm.SVC(kernel='linear', C=1000); clf.fit(X, y)`

C: `clf = svm.SVC(kernel="rbf", C=1, gamma=1); clf.fit(X, y)`

D: `clf = svm.SVC(kernel="rbf", C=1, gamma=10); clf.fit(X, y)`

E: `clf = svm.SVC(kernel="euclidean", C=1, gamma=2); clf.fit(X, y)`

F: `clf = svm.SVC(kernel="euclidean", C=1000, gamma=2); clf.fit(X, y)`

G: `clf = svm.SVC(kernel="gini", C=1); clf.fit(X, y)`

H: `clf = svm.SVC(kernel="gini", C=1000); clf.fit(X, y)`

i. _____ The data are two linearly-separable clouds of points, one red and one blue.
ANSWER: B

ii. _____ The red points are scattered between $x = -3$ and $x = 3$ and roughly along $y = x^2$, a parabola with vertex $(0, 0)$ that opens up. The blue points are scattered between $x = -3$ and $x = 3$ and roughly along $y = x^2 + 2$, a parabola with vertex $(0, 2)$ that opens up.
ANSWER: C

iii. _____ The data consist of two clouds of points, one red and one blue, that are linearly-separable except for a few outliers of each color.
ANSWER: A

iv. _____ The data consist of mixed red and blue points scattered randomly in the region $x \in [0, 1]$ and $y \in [0, 1]$.
ANSWER: D

4. Consider finding the linear regression line by hand for the points $(1,3),(2,5),(3,4)$. Match each mathematical quantity on the left with its value on the right.

(a) $\mathbf{X} =$ _____ G: must by 3×2

(b) $\mathbf{X}^T\mathbf{X} =$ _____ B: must be 2×2

(c) $\mathbf{y} =$ _____ D: from data

(d) $\mathbf{w} =$ _____ C: must be 2×1

(e) $\hat{\mathbf{y}} =$ _____ E: must be 3×1 and D is taken by $\mathbf{y}$

(f) $f_{\mathbf{w},b}(3) =$ _____ A: must be scalar and 0.5 is taken by b

(g) intercept = _____ F: first element of $\mathbf{w}$

(My answers above are mostly from considering dimensions. With a little more work, the matrix arithmetic can be done to find the answers.)

(A) 4.5

(B) $\begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix}$

(C) $\begin{bmatrix} 3 \\ 0.5 \end{bmatrix}$

(D) $\begin{bmatrix} 3 \\ 5 \\ 4 \end{bmatrix}$

(E) $\begin{bmatrix} 3.5 \\ 4.0 \\ 4.5 \end{bmatrix}$

(F) 3

(G) $\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$

5. Consider 3-NN (three nearest neighbors) using the Minkowski distance with $p = 1$.

(a) Find the distance from $\mathbf{z} = (3,4)$ to each of the other points $\mathbf{x}$:

$\mathbf{x}$	y	Distance from $\mathbf{z}$ to $\mathbf{x}$
$(-1,-1)$	1	ANSWER: 9
(0, 1)	0	ANSWER: 6
(1, 0)	0	ANSWER: 6
$(2,3)$	1	ANSWER: 2

(b) How does 3-NN classifiy $\mathbf{z}$?
ANSWER: 0
The 3-NN have distances 6, 6, and 2 and y values 0, 0, and 1. 0 is most frequent.

(c) What y value does 3-NN regression predict for $\mathbf{z}$?
ANSWER: $\hat{y} = \frac{1}{3}(0+0+1) = \frac{1}{3}$

(d) How does weighted 3-NN classifiy $\mathbf{z}$?
ANSWER: 1
The reciprocals of the 3 smallest distances are $\{d_i\}$ are $\frac{1}{6}$, $\frac{1}{6}$, and $\frac{1}{2}$. The weight on $\hat{y} = 0$ is $\frac{1}{6} + \frac{1}{6} = \frac{1}{3}$. The weight on $\hat{y} = 1$ is $\frac{1}{2}$.

6. Suppose we have run this code, which reads the `data_string` into a data frame `df` and then creates `X` and `y` from `df`:

```
from io import StringIO
import pandas as pd

data_string = """
x1,     x2,  x3,  x4,  y
0,   -0.01,   1,  13,  5
1,   -0.00,   0,  12,  6
2,   -0.03,   3,  11,  7
3,   -0.02,   2,  10,  8
"""
df = pd.read_csv(StringIO(data_string), sep='\s*,\s+', engine='python')
X = df[['x1', 'x2', 'x3', 'x4']]
y = df.y
```

Hint: I answered these questions without doing calculations. I just inspected the data.

(a) The following code prints some subset of the data.

```
from sklearn.feature_selection import VarianceThreshold, SelectKBest
from sklearn.feature_selection import r_regression, f_regression

selector = VarianceThreshold(threshold=0.1)
selector.fit_transform(X)
```

Circle the names of the features that are displayed by the last line:

`x1,` `x2,` `x3,` `x4,` `y`

ANSWER: `x1,` `x3,` `x4`

(Feature `x2` has standard deviation less than its range of 0.03, and variance less than that, and was excluded.The other features have variances much higher than the `threshold`.)

(b) The following code prints some subset of the data.

```
selector = SelectKBest(score_func=r_regression, k=2)
selector.fit_transform(X, y)
```

Circle the names of the features that are displayed by the last line:

`x1,` `x2,` `x3,` `x4,` `y`

ANSWER: `x1,` `x3`

(These are the features with positive correlation r with `y`, so they are the two with highest r. Note that "highest" means "closest to 1," not "closest to ± 1.")

(c) The following code prints some subset of the data.

```
selector = SelectKBest(score_func=f_regression, k=2)
selector.fit_transform(X, y)
```

Circle the names of the features that are displayed by the last line:

`x1,  x2,  x3,  x4,  y`

ANSWER: `x1,  x4`

(These are the features with r nearest ± 1 and therefore highest $F = \frac{R^2}{1-R^2}(n-2)$.)

7. Mark each statement TRUE or FALSE by circling the appropriate choice.

(a) TRUE / FALSE In linear regression, a reasonable alternative to the cost function *mean squared error* $= \frac{1}{N}\sum_{i=1}^{N}[f_{\mathbf{w},b}(\mathbf{x}_i) - y_i]^2$ is *sum of squared error* $= \sum_{i=1}^{N}[f_{\mathbf{w},b}(\mathbf{x}_i) - y_i]^2$.

ANSWER: TRUE. The two functions will have the same minimum, as they differ only by the constant $\frac{1}{N}$.

(b) TRUE / FALSE Using gradient descent to minimize $z = f(\mathbf{x})$ can be slow because the algorithm requires many calls to f, which is expensive if N (number of examples) is large or D (number of features) is large.

ANSWER: FALSE. Gradient descent makes no calls to f. It can make many calls to the gradient ∇f.

(c) TRUE / FALSE For the soft-margin SVM with decision boundary $\mathbf{w}\mathbf{x} + b = 0$ where $\mathbf{w} = (1, 2)$ and $b = 3$, the example $(\mathbf{x}, y) = ((4, 5), -1)$ has hinge loss 18.

ANSWER: TRUE, as hinge loss is $\max(0, 1 - y_i(\mathbf{w}\mathbf{x} + b)) = \max(0, 1 - (-1)((1 \cdot 4 + 2 \cdot 5) + 3)) = 18$.

(d) TRUE / FALSE For training data $\{(\mathbf{x}, y)\}$ such that $\mathbf{x}_i \neq \mathbf{x}_j$ for all i and j, we can build a 3NN model that classifies the training examples without error.

ANSWER: FALSE. For example, if there are only $N = 3$ examples with $y_1 = 0, y_2 = 0, y_3 = 1$, all three examples be classified as the majority y value, namely 0.

(e) TRUE / FALSE If we train a hard-margin linear SVM on linearly separable data, then discard training examples which are support vectors, and then train a new SVM on the remaining examples, the first SVM will have a wider "road" than the second.

ANSWER: FALSE. The first "road" will be narrower than the second.

(f) TRUE / FALSE A linear SVM with decision boundary $(1, 2, 2) \cdot \mathbf{x} - 2 = 0$ has a smaller margin between $+1$ and -1 support vectors than one with boundary $(1, 4, 8) \cdot \mathbf{x} + 3 = 0$.

ANSWER: FALSE. The margin for the first SVM is $\frac{2}{||\mathbf{w}||} = \frac{2}{\sqrt{1^2+2^2+2^2}} = \frac{2}{3}$, while the margin for the second is $\frac{2}{||\mathbf{w}||} = \frac{2}{\sqrt{1^2+4^2+8^2}} = \frac{2}{9}$.

(g) TRUE / FALSE Every decision tree regression function is a step function.

Hint: A *step function* is a function that is constant over each of one or more intervals.

ANSWER: TRUE.

(h) TRUE / FALSE Every k-NN regression function is a step function.

Hint: A *step function* is a function that is constant over each of one or more intervals.

ANSWER: TRUE.

(i) TRUE / FALSE In logistic regression, we use the natural log function to facilitate finding a closed-form expression for the coefficients $\mathbf{w}$ and b in terms of the data.

ANSWER: FALSE. We don't have such a closed-form expression.

(j) TRUE / FALSE Gradient descent can fail to converge to a global minimum if it gets stuck in a local minimum.

ANSWER: TRUE.

8. Here are some questions on feature engineering.

(a) Do min-max rescaling on feature `x`:

(input) `x`	(output) `x_rescaled`
1	
0	
3	

ANSWER:

(input) `x`	(output) `x_rescaled`
1	$\frac{1}{3}$
0	0
3	1

(b) Use one-hot encoding to transform the categorical feature `weather` into binary features with reasonable names.

(input) `weather`	(output)
cloudy	
rainy	
sunny	
cloudy	

ANSWER:

(input)	(output)		
`weather`	`cloudy`	`rainy`	`sunny`
cloudy	1	0	0
rainy	0	1	0
sunny	0	0	1
cloudy	1	0	0

9. Here are some questions about decision trees.

(a) Consider a classification decision tree node containing the set of examples $S = \{(\mathbf{x}, y)\}$ where $\mathbf{x} = (x_1, x_2)$:

S

x_1	x_2	x_3	y
2	11	12	0
1	8	14	1
0	6	17	1
3	10	15	0
4	7	16	1
5	9	13	0

i. The entropy of this node in bits is _____.

ANSWER:

The node's y values are 0, 1, 1, 0, 1, 0, so $f_{ID3}(S) = \frac{1}{|S|}\sum_{(\mathbf{x},y)\in S} y = \frac{1}{6}(0+1+1+0+1+0) = \frac{1}{2}$.

$H(S) = -\frac{1}{2}(-1) - \frac{1}{2}(-1) = 1$

(Or, since a random draw from S amounts to a coin flip, the entropy is 1 bit.)

ii. The (feature, threshold) pair (j, t) that yields the best split for this node is feature $j =$ _____ and threshold $t =$ _____.

ANSWER:

Using feature $j = 2$ and threshold $t = 8.5$ (or any $t \in [8, 9)$) splits S into $S_- = \{(\mathbf{x}, y) \in S | x^{(j)} \leq t\}$ and its complement $S_+ = \{(\mathbf{x}, y) \in S | x^{(j)} > t\}$, each of which has entropy 0.

(b) Consider a regression decision tree with `max_depth=1` (that is, the root node is split once into two leaves) made from the set of examples $S = \{(\mathbf{x}, y)\}$ where $\mathbf{x} = x_1$:

S

x_1	y
0	1
1	2
2	11
3	12
4	13
5	14

What value does this tree predict for $x_1 = 2.5$? _____

ANSWER: 12.5

The best split uses feature $j = 1$ and threshold $t = 1.5$, yielding a left subtree containing the first two examples and a right subtree containing the last four examples. Making a predition with $x_1 = 2.5$ would use the right subtree. Its average y is 12.5, so the tree would predict $\hat{y} = 12.5$.