DATA 301 Fall 2025, Midterm Exam Key

October 21th, 2025

Allocated Time: 75 minutes

Name: _____ Banner ID:_____

By signing my name above, I affirm the below pledge:

> *"As a member of the William & Mary community, I pledge on my honor not to lie, cheat, or steal, either in my academic or personal life. I understand that such acts violate the Honor Code and undermine the community of trust, of which we are all stewards."*

DO NOT OPEN THE EXAM UNTIL INSTRUCTED TO BEGIN

- This is a closed-book exam. You may use one standard 8.5" by 11" piece of paper with any notes you deem appropriate or significant (front and back). No devices connected to the Internet are allowed. Do not communicate with any other person.
- Choose the correct answers to the questions and fill out the Answer Sheet.
- Suggestion: Read the entire exam before starting work on any problem. Budget your time wisely.
- Make sure you have the name written on all pages.
- This is the in-class part of the final exam, which consists of 15 questions. The coding part of the midterm must be completed and submitted on Blackboard by 11:59 pm (October 21).
- Partial credit will be considered.
- Every question is weighted proportionally to its number of possible answers. The whole test is worth 100 points overall.
- Good luck!

Multiple Answer Questions. Read all possible answers and choose only the correct ones.

Question 1: Levels of Measurement

A researcher collects the following variables in a study: (1) temperature in Kelvin, (2) education level (high school, bachelor's, master's, PhD), (3) Social Security Numbers, (4) time to complete a task in seconds, and (5) Date of the Year. Which statements are correct?

A) Temperature in Kelvin is a Ratio-level variable because it has a true zero point.

B) Education level is that an Interval variable because we can measure the distance between degree levels.

C) Social Security Numbers are Nominal-level variables with no relative numerical meaning.

D) Calculating the ratio "twice as much" is meaningful for both temperature in Kelvin and time in seconds.

E) Dates are Ordinal variable they have an order without meaningful intervals.

F) Both Social Security Numbers and Education Level are Ordinal Data with a distinct hierarchy.

Question 2: Conditional Probability and Bayes' Theorem

A new medical test for a certain disease has been developed. The disease affects 1 in 1000 people. The test has a 95% true positive rate and a 98% true negative rate. Which of the following statements are correct?

A) The prior probability of having the disease is 0.1%.

B) Bayes' Theorem works best on independent events to calculate probabilities.

C) The false positive rate is 5%.

D) If a person tests negative, they are more likely not to have the disease than to have it.

E) The amount of people who got a positive test, but do not have the disease is about 2%.

F) The probability of a person having the disease given a positive test result is less than 5%.

Question 3: Classes, Dunder Methods, and Scaling

You are implementing a custom StandardScaler class for normalizing features in a machine learning pipeline. Consider the following code structure:

```python
class StandardScaler:
    def __init__(self):
        self.mean_ = None
        self.std_ = None

    def fit(self, X):
        self.mean_ = X.mean(axis=0)
        self.std_ = X.std(axis=0)
        return self

    def transform(self, X):
        return (X - self.mean_) / self.std_
```

Which of the following statements are true about implementing and using this scaler correctly?

A) Methods in a class can be called 'before' their definition (such as calling the transform() method inside of the fit() method).

B) You should fit the scaler separately on both training and test sets to ensure each dataset is properly normalized

C) There are no circumstances in which you would want to scale your target data (y).

D) The scaler should be fit only on the training data, then the same mean and std values should be used to transform both training and test sets.

E) Improper scaling will only ever lead to worse performance in loss metrics on your testing data.

F) Dunder Methods are special methods which run automatically in certain contexts, such as enabling the behavior of the instantiated to interact seamlessly with Python's built-in functions, operators, and language constructs, without needing to be explicitly called.

Question 4: Monte Carlo Simulations

Consider using a Monte Carlo simulation to estimate the average value for the longest chain of continuous heads when flipping 100 coins using N iterations. Which of the following statements are true?

A) The Law of Large Numbers provides the theoretical foundation for why this Monte Carlo method works.

B) A Monte Carlo simulation will likely result in exactly the true value when $N = 1,000$.

C) If we use $10,000$ iterations, the error in our estimate will be approximately $\frac{1}{10}$th the error with $1,000$ iterations.

D) This method is only suitable when there is an analytical solution for this result.

E) For this specific problem, the Monte Carlo simulation will exhibit strong convergence.

F) Assuming a constant value of N, flipping $1,000$ coins instead of 100 coins would result in the standard deviation being 10 times higher.

Question 5: Central Limit Theorem and Confidence Intervals

Which of the following statements about the Central Limit Theorem (CLT) and confidence intervals are correct? (Select all that apply)

A) The Central Limit Theorem is applicable even if the sample size is small (e.g., $n < 10$), as long as the underlying population is normally distributed.

B) A 99% confidence interval for a population mean will contain the 95% confidence interval for the same population mean, assuming the same sample size.

C) The Central Limit Theorem states that the distribution of the population itself will become approximately normal as the sample size increases.

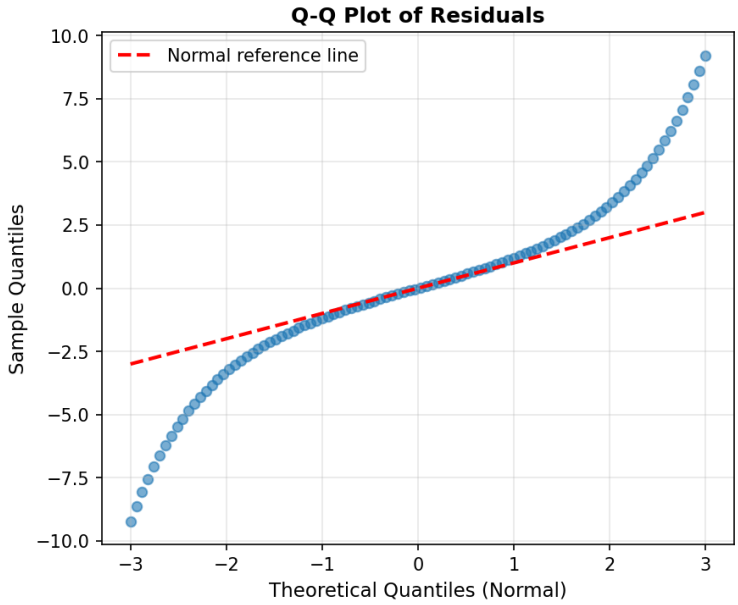
D) For a non-normally distributed population, the distribution of sample means will be exactly normal for any sample size greater than 30.

E) The probability that a single specific value from a continuous uniform distribution [0, 10] is exactly 5 is 0.1.

F) The margin of error in a confidence interval for a population mean decreases as the sample size increases.

Question 6: Residual Diagnostics and Q-Q Plots

After fitting a linear regression model, you examine the residuals to assess model assumptions. The Q-Q plot is shown below:



Based on the Q-Q plot and general principles of residual analysis, which statements are true?

A) The Q-Q plot shows that the residuals deviate from normality primarily in the tails, suggesting heavier tails than a normal distribution.

B) If the Q-Q plot shows all points exactly on the reference line, this suggests the residuals are approximately normally distributed.

C) We would expect the distribution of the residuals to be more spread out and uniform than the expected normal distribution.

D) The Anderson-Darling (AD) test is more sensitive to departures from normality in the tails compared to the Kolmogorov-Smirnov (KS) test.

E) Residuals that are not normally distributed provide no indication that the linear regression model is inappropriate.

F) A distribution that is skewed left/right relative to a normal distribution would produce an S-shaped Q-Q plot as above.

Question 7: Profiling, Runtime Usage Calculation, & Scaling

You are provided with the following code and profiling output (on the next page):

```
1  #Class Definition
2  class RandomWalk:
3      def __init__(self):
4          pass
5
6      def sim_population(self, num_walks, num_steps, num_dims):
7          #Simulate the random walk moves
8          moves = np.random.rand(num_walks, num_steps, num_dims)
9          positions = np.cumsum(moves, axis=1)
10         return positions
11
12     def compute_radius(self, positions):
13         #Compute the radius from the origin
14         radii = np.linalg.norm(positions, axis=2)
15         return radii
16
17     def final_radius_plot(self, dims, num_walks, num_steps):
18         #Simulate the random walk and compute the average final radius values for many dims
19         avg_radii = []
20         for dim in dims:
21             positions = self.sim_population(num_walks, num_steps, dim)
22             radii = self.compute_radius(positions)
23             final_radii = radii[:, -1]
24             avg_radius = np.mean(final_radii)
25             avg_radii.append(avg_radius)
26
27         #Plot the average final radius vs dimensions
28         plt.figure(figsize=(10, 6))
29         plt.plot(dims, avg_radii, marker='.', c='red')
30         plt.xlabel('Number of Dimensions')
31         plt.ylabel('Average Final Radius')
32         plt.title('Average Final Radius vs Number of Dimensions in Random Walk')
33         plt.grid()
34
35 #Example usage
36 def func():
37     RW = RandomWalk()
38     dims = np.arange(1, 51, 1)
39     RW.final_radius_plot(dims, num_walks=1000, num_steps=1000)
```

Based on these figures, which of the following statements are correct?

A) line 21 is the correct line to identify when diagnosing where most of our runtime is spent.

B) To calculate the distribution of runtime among lines, the easiest way is to use the 'Time' results of profiling outputs.

C) We are calculating $51,000,000$ different positions.

D) line 8 takes up $\approx 50\%$ of the total runtime for this code.

E) Leaving everything else constant, an increase in the Per-Hit time taken of a line will result in a matching additive percentage increase in the % Time of that line.

F) numpy vector operations (such as np.mean) will run much faster than 'for loops', especially when the number of iterations is high.

```
Timer unit: 1e-07 s

Total time: 11.9839 s
File: C:\Users\Owner\AppData\Local\Temp\ipykernel_3288\3773468216.py
Function: RandomWalk.sim_population at line 6

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     6                                           def sim_population(self, num_walks, num_steps, num_dims):
     7                                               #Simulate the random walk moves
     8        50   72429952.0 1.45e+06     60.4          moves = np.random.rand(num_walks, num_steps, num_dims)
     9        50   47409249.0 948185.0     39.6          positions = np.cumsum(moves, axis=1)
    10        50        274.0      5.5      0.0          return positions

Total time: 2.40163 s
File: C:\Users\Owner\AppData\Local\Temp\ipykernel_3288\3773468216.py
Function: RandomWalk.compute_radius at line 12

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    12                                           def compute_radius(self, positions):
    13                                               #Compute the radius from the origin
    14        50   24015795.0 480315.9    100.0          radii = np.linalg.norm(positions, axis=2)
    15        50        491.0      9.8      0.0          return radii

Total time: 14.7843 s
File: C:\Users\Owner\AppData\Local\Temp\ipykernel_3288\3773468216.py
Function: RandomWalk.final_radius_plot at line 17

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    17                                           def final_radius_plot(self, dims, num_walks, num_steps):
    18                                               #Simulate the random walk and compute the average final radius values for many dims
    19         1          4.0      4.0      0.0          avg_radii = []
    20        51        672.0     13.2      0.0          for dim in dims:
    21        50  123549785.0 2.47e+06     83.6              positions = self.sim_population(num_walks, num_steps, dim)
    22        50   24019650.0 480393.0     16.2              radii = self.compute_radius(positions)
    23        50      84014.0   1680.3      0.1              final_radii = radii[:, -1]
    24        50      30775.0    615.5      0.0              avg_radius = np.mean(final_radii)
    25        50        514.0     10.3      0.0              avg_radii.append(avg_radius)
    26
    27                                               #Plot the average final radius vs dimensions
    28         1       9408.0   9408.0      0.0          plt.figure(figsize=(10, 6))
    29         1     139890.0 139890.0      0.1          plt.plot(dims, avg_radii, marker='.', c='red')
    30         1        787.0    787.0      0.0          plt.xlabel('Number of Dimensions')
    31         1        671.0    671.0      0.0          plt.ylabel('Average Final Radius')
    32         1       3173.0   3173.0      0.0          plt.title('Average Final Radius vs Number of Dimensions in Random Walk')
    33         1       3846.0   3846.0      0.0          plt.grid()
Total time: 14.792 s
File: C:\Users\Owner\AppData\Local\Temp\ipykernel_3288\3773468216.py
Function: func at line 36

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    36                                           def func():
    37         1         29.0     29.0      0.0          RW = RandomWalk()
    38         1        144.0    144.0      0.0          dims = np.arange(1, 51, 1)
    39         1  147919406.0 1.48e+08    100.0          RW.final_radius_plot(dims, num_walks=1000, num_steps=1000)
```

Question 8: Reinforcement Learning and MDPs

In the context of reinforcement learning, specifically Markov Decision Processes (MDPs), which statements are true?

A) Markov Decision Processes are a framework to solve RL problems, any problem can be formulated into an MDP as there are no required properties.

B) A stochastic policy always chooses the same action in a given state with probability 1.

C) A policy in a Markov Decision Process defines which action to take in each state, regardless of the rewards received by the agent.

D) UCB will always converge on the optimal policy if properly configured. (With infinite runtime)

E) Epsilon Greedy will always converge on the optimal policy if properly configured. (With infinite runtime)

F) The discount factor, γ, balances the importance of immediate rewards versus future rewards.

Question 9: Strategies for the Multi-Armed Bandit Problem

An online platform wants to test 5 different advertisements (bandits) to see which one has the highest click-through rate. Which of the following strategies and concepts are correctly described?

A) An epsilon-greedy strategy with $\epsilon = 0$ will always choose the action with the current highest estimated reward.

B) The Upper Confidence Bound (UCB) algorithm is generally more effective than a simple greedy approach because it accounts for uncertainty in the estimated rewards.

C) A greedy strategy (pure exploitation) guarantees finding the optimal advertisement as long as each arm is tried at least once initially.

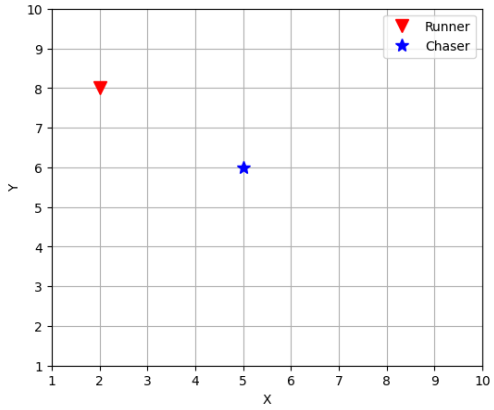
D) With fewer possible actions, UCB has a higher chance of failing to converge.

E) The exploration rate in an epsilon-greedy strategy should increase over time as we collect more data to ensure we continue discovering better options.

F) We need to properly understand each advertisement strategy in order to formulate a RL framework to select the optimal one.

Question 10: MDP Terminology & Setup

Consider the following MDP problem: You have two agents on a $10 * 10$ grid. The 'Chaser' agent will always pick some direction in which gets it closer to the 'Runner' agent and step the maximum distance. You want to use MDP frameworks to train the 'Runner' agent to evade the 'Chaser' agent. Neither agent can go to any position not on the grid. Both agents can move a total distance $dx + dy \leq 2$ each step.



Which of the following statements are true?

A) A valid State for this problem would be the 'Runner' agent's current (x, y) position.

B) The 'Runner' agent's current velocity is a required component of the State.

C) This problem has approximately $10,000$ possible States.

D) UCB is better suited for a task like this than Epsilon Greedy.

E) There are approximately $1,440,000$ possible State-Action Pairs.

F) We will likely want to implement multiple type of rewards/penalties.

Question 11: Gradient Descent Optimization Algorithms

Compare the following gradient descent variants: standard (basic) gradient descent, mini-batch stochastic gradient descent (SGD), RMSProp, and Adam. Which statements are correct?

A) Mini-batch SGD typically converges faster than basic gradient descent per epoch because it updates parameters more frequently.

B) RMSProp uses momentum to accelerate convergence but does not adapt the learning rate for individual parameters.

C) Adam combines ideas from both momentum-based methods and RMSProp by maintaining both integrating a momentum and accumulation term.

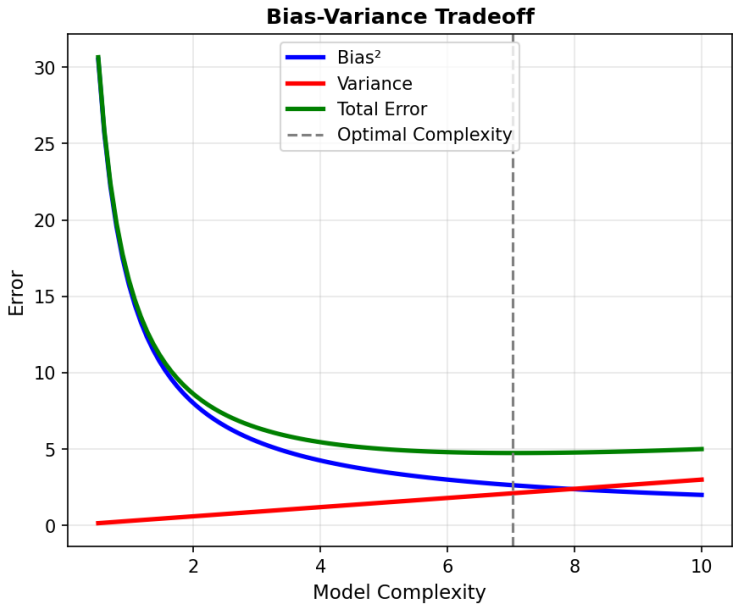
D) Basic gradient descent always finds the global minimum for non-convex (not continuously curving up at all points) loss functions.

E) Mini-batch SGD introduces noise into the optimization process through its selection of batches, which can help escape local minima.

F) Being trapped in a Local Minima is the only situation in which an optimizer will fail to converge onto the Global Minimum.

Question 12: Bias-Variance Tradeoff

The figure below illustrates the bias-variance tradeoff in model complexity:



Select all correct statements about this tradeoff:

A) The optimal model complexity is at the point where bias equals variance.

B) Variance is a measure of how much a model 'learns' the noise present in the training data.

C) Overfitting occurs when a model has low bias and low variance.

D) A model with high variance is likely to perform similarly well on both training and test data.

E) Regularization techniques such as Ridge and Lasso help reduce variance at the cost of increased bias.

F) Underfitting is characterized by high bias and low variance.

G) Total Error remains flat in a model as the Complexity goes toward infinity.

Question 13: Regularization: Lasso, Ridge, and Elastic Net

Consider training a linear regression model with 100 features, where only 10 features are truly relevant and many features are highly correlated and redundant. Compare Lasso (L1), Ridge (L2), and Elastic Net regularization:

A) Lasso regression can produce sparse models by setting some coefficients exactly to zero, effectively performing feature selection.

B) When features are highly correlated, Lasso tends to distribute weights evenly among all correlated features.

C) Elastic Net combines L1 and L2 penalties and can handle correlated features better than Lasso alone.

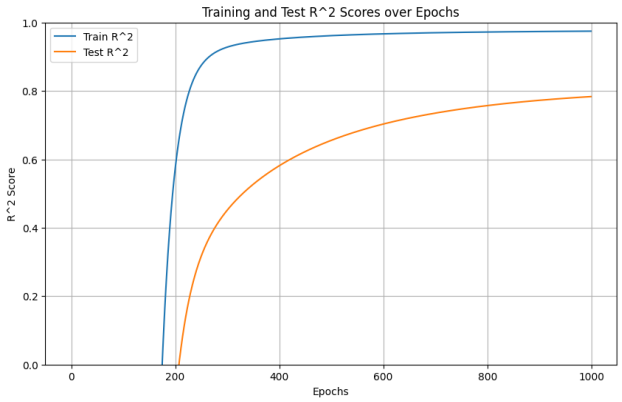
D) Ridge Regression does not select features and thus does not reduce model complexity.

E) Ridge regression will shrink coefficients toward zero but will not set them exactly to zero.

F) The regularization parameter α controls the strength of the penalty, with smaller values leading to more regularization.

Question 14: Overfitting and Underfitting

You have fit a Linear Model to your data and plot your training and testing scores over time:



Which of the following statements are true?

A) The model is overfitting, performing significantly better on Training Data than Testing Data.

B) We can tell the model is Underfitting as the Testing Score is still increasing.

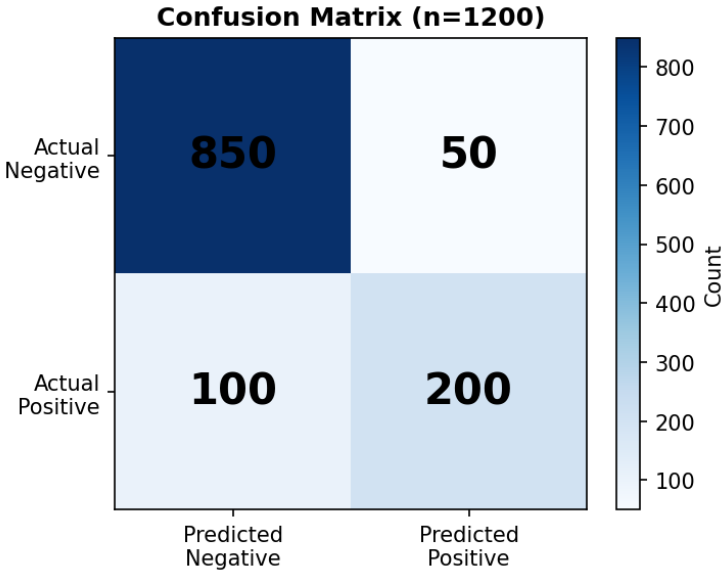
C) Decreasing the magnitude of each weight a small amount each epoch would have similar effects as Regularization.

D) Training for longer would eventually close the gap between the Training and Testing Scores.

E) A model with better performance on Training Data than Testing data can always be considered overfit and should be addressed.

Question 15: Classification Metrics and Imbalanced Datasets

You are building a binary classifier for a rare disease where only 2% of patients have the disease. The confusion matrix for your model on a test set of 1,200 patients is shown below:



Select all correct statements:

A) The accuracy of this model is approximately 87.5%.

B) We can compensate for any level of class imbalance by changing our scoring metric to weight the minority classes more.

C) The precision (positive predictive value) is approximately 50.0%.

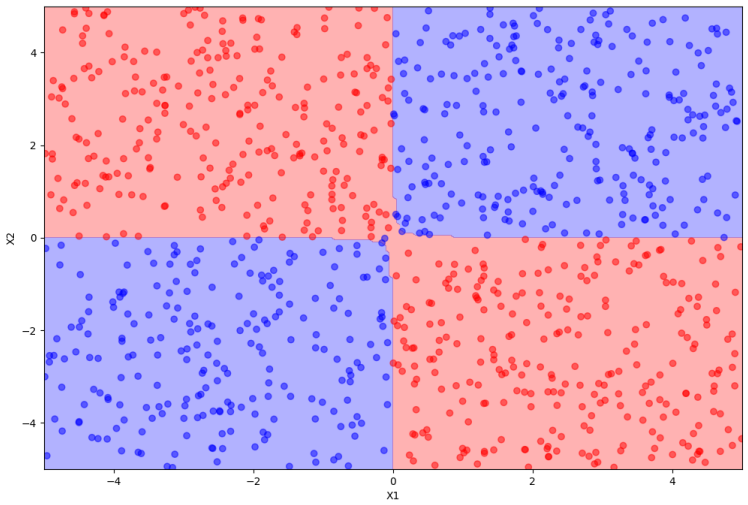
D) The recall (sensitivity/true positive rate) is approximately 66.7%.

E) For this imbalanced dataset, accuracy is a misleading metric because a naive classifier that always predicts "negative" would achieve $\approx 80\%$ accuracy.

F) The F1-score does not care which class is considered the primary class.

Question 16: Classification Decision Boundaries

The figure below shows a classification task and the approximate decision boundaries:



Which statements about this task are correct?

A) KNN is likely to produce this decision boundary.

B) A Random Forest model is well-suited to this data.

C) A Logistic Regression model with $X1$ and $X2$ as inputs would perform well on this data.

D) A SVM model with the 'sigmoid' kernel would be well suited for this data.

E) A Logistic Regression model with $X1 * X2$ as input would perform well on this data.

END OF THE EXAM