**CSCI 5622 ML Exam (Fall 2023)**

**Honor Statement:**

**I, \_\_Vaishnavi Asuri\_\_, understand that I will do this exam alone and without the help of other humans. I understand exactly what it means to “do my own work” and I will do my own work. If I fail to do this and/or my work looks similar to other work, I understand that very bad things can happen.**

**Signature \_\_\_VA\_\_\_Date\_2/9/23\_**

**Directions: Please complete this Exam using THIS DOCUMENT.**

**Download and save this document.** Place all answers, work, illustrations, images, etc. that you want graded into this document.

If you are asked to code, you can create you code in Jupyter, as .py, on git, or whatever. You just need to have a link to your code that you will place on this document. Therefore, when you submit this exam, you will submit only this document.

**Please save this document as: YourName\_ML\_Exam\_2023.docx.**

**Notes and Rules:**

1. **No questions are permitted** by anyone for any reason during this exam. Follow the instructions, do not overcomplicate things, and make assumptions (that you clearly write down) if/as needed. Part of the test is your ability to “do the test”.
2. **It is not permitted to work with any other humans** on this Exam. While I am not concerned about this as I feel that all of you are very ethical, I am required to note that anyone who works together must get a “0” grade and may both fail the class and potentially have further issues with the program. Just do your own work 😊
3. **This Exam is open** – meaning you can use the web, class notes, my website, my code, your code, etc. **If you are in doubt about something – do not use it**. For example, if your buddy Bob posts code on the web that answers one of these questions and then you use it – that’s cheating. Using my code or your code is fine.
4. **This Exam will be due no later than 12/10 (Sunday) by 11:59 pm MT**. If you wait until the last hour to submit and run into a problem, there will be no solution. This Exam cannot be late for any reason. Please do not test this. Submit EARLY – like 24 hours early! I have set up the submission area so that you can **submit as many times as you want.** So, you can submit in advance (like Saturday), then you can do more work on Sunday if you wish, and then you can submit again if you want. **We will use the LAST (latest) submission** for grading. This way, there is no need to wait until the last minute to submit. Again, to be clear, the **system will close and will lock at 11:59pm MT on 12/10. Exams not submitted by that time will not be graded.** Also – for those of you who always push things – if you submit the “wrong version” or a “poem for your honey” instead of the Exam – that will **not matter!** **BE CAREFUL**. **Submit early and submit correctly.** (It makes me sad that I have to say all of this 😐)

**Part 1: The Interview**.

To answer the following questions, pretend you are on an interview. Interviewers are generally looking for brief answers (at least at first) and are trying to see what you do and do not know. Writing “more” does not show that you know more – and often – it suggests that you cannot articulate your answer succinctly.

Therefore, answer each of the following questions using 2 – 4 sentences (never more than 4). Be precise, concise, and succinct. Each topic here has books written about them. I am not looking for a book – just a few sentences that directly, quickly, and clearly answer the question. (All questions below were taken from interviews)

1. **What is Ensemble Learning and give an example?**

Ensemble Learning enhances machine learning by combining multiple models for more accurate predictions. The concept used is that multiple models perform better as they compensate for each other’s shortcomings. If we use a movies dataset, Ensemble Learning integrates Naive Bayes, KNN, and Convolutional Neural Network models to predict actors, leveraging their probabilistic, semantic, and visual strengths for well-rounded predictions and overcoming individual model limitations.

1. Briefly explain the difference between supervised and unsupervised learning?

Supervised learning involves labeled data, predicting specific outcomes with known labels, while unsupervised learning uses unlabeled data to discover inherent patterns. Supervised learns from existing labeled data and predicts outcomes, e.g., classification. Unsupervised finds patterns, e.g., clustering, without labels. Evaluation differs, with supervised focusing on prediction accuracy, and unsupervised on outcome interpretability.

1. What is a “Training set” and a “Testing Set” in supervised learning and how are they used?

The training set, comprising 60%-80% labeled data, maps inputs to known target variables, like movie plot descriptions to starring actors. The supervised learning model uses this set to infer connections inductively, tuning patterns incrementally. The testing set, 20%-30% unseen data, assesses the model's predictive accuracy, ensuring unbiased generalization to new data.

1. How would you handle or manage missing or incorrect data in a record dataset?

To manage missing or incorrect data in my record dataset, I identify the extent and patterns of missing data. For small issues, I omit rows; for larger gaps, I use imputation models like Mean, Median, or predictive methods. Advanced techniques like multiple imputation require careful consideration. Regular audits ensure prompt handling of data issues.

1. **In a confusion matrix, what is a false negative, and give an example where a false negative could cause a very serious issue.**

In a confusion matrix, False Negatives occur when the model wrongly predicts a positive outcome as negative, potentially with serious consequences. In scenarios like detecting deadly viruses (like Covid 19) or fast-progressing diseases(like Cancer), a false positive means undetected threats, leading to health degradation. Delayed diagnosis in communicable diseases results in rapid spread due to misinformation.

1. **Suppose you have a labeled dataset where the labels are one of three categories (like dog, cat, mouse). However, suppose that 90% of the data is one label (like cat) and 10% of the data are the other two labels (like dog and mouse). Describe the issue here and a possible resolution.**

Imbalanced datasets, where one class dominates, can lead to biased results and poor performance in minority classes. Techniques to address this include data augmentation, adjusting class weights in the loss function to focus on underrepresented groups, and resampling by undersampling or oversampling to balance class occurrences for more effective model training.

1. **Describe how supervised learning can be used to detect email spam. You describe this – this question is not asking you to paste from the web 😊**

In spam filtering, I leverage supervised Machine Learning algorithms for classification. First, I gather labeled data, assembling a diverse dataset of emails categorized as "spam" or "wanted." After light data cleaning, I numerically represent features like links and trigger words. Support Vector Machines and other algorithms learn rules from these features for accurate email categorization. Continuous monitoring and retraining adapt to evolving spam tactics, keeping the system effective.

1. **Name two unsupervised learning methods and briefly describe what they do.**

Unsupervised learning involves predicting underlying patterns without labeled data, achieved through methods like Association, clustering, and dimensionality reduction. K Means Clustering divides data into K clusters based on centroids, minimizing within-cluster variance. Principle Component Analysis identifies directions of maximum variance, simplifying complex data structures to reveal core relationships and underlying patterns by focusing on the most meaningful dimensions.

1. **Name three supervised learning methods and briefly describe what they do.**

Three supervised learning methods include Decision Trees, which recursively split data based on features to form a tree structure for classification or regression. Linear Regression predicts outcomes using predictor variables, establishing a linear relationship. Support Vector Machines draw a hyperplane to efficiently separate data into categories, maximizing margin in linear separations or using a kernel trick for complex cases.

1. **Suppose you work for a store and you want to recommend items to customers based on past purchases. Which machine learning method would you use? Describe an example.**

Say, this is a Bookstore. As a bookstore employee, I'd utilize customer data, browsing history, and demographic info with a decision tree algorithm. This would segment customers based on genre preferences, author followings, and more, creating profiles like "mystery & and thriller readers." The decision tree's interpretability allows me to tailor recommendations based on precise reasoning from historical data analysis.

**Part 2: SVMs**

**Question 1:** Here, you will illustrate the steps needed to solve for the Lagrange multipliers and then for the equation (w and b) of the SVM separating line. You may refer to the SVM slides for an example. Show all of your work. Do this right here in Word on this document. There is no need to write things out by hand 😊 If you must, you can use the “draw” tool in Word - but only as needed for images.

1. Choose any **three** data points that are 2D. (Example, the point (2,3) is a 2D datapoint) [do not use the same values that I use on the slides in class 😉]

Chosen Points:

X1= (1,0) X2= (2,2) X3= (5,6)

Y1=1 Y2= -1 Y3=1

1. Create a plot (you can use “draw” for this) that shows the cartesian coordinate system, the three points, and their labels as +1 or -1. You can choose the labels as long as you represent both labels.

A graph with points and labels

Description automatically generated with medium confidence

1. Your next goal is to solve for the SVM model – which you will need **w** and b for.

To calculate ‘w’ and ‘b’, we need the value of λi (Lagrange multipliers) which is obtained from this equation:

w = λi yi xi = (λ1 \* y1 \* x1) +(λ2 \* y2 \* x2) + (λ3 \* y3 \* x3)

yi(wTxi + b) – 1 = 0

1. Show all of your work and steps to do this. When you are done, draw your calculated separator line (your SVM model) onto the cartesian coordinate system so illustrate that it does separate your points correctly.

To calculate above we need to follow these steps. So let us break it down-

1. the dot products of X1, X2 and X3
2. Substitute the dot products and the labels (y values) in: L(λ) = i – ½
3. Calculate the weighted vector w and then b

**Step a:**

x1= (3,0) x2= (1,2) x3= (0,1)

y1= 1 y2=1 y3=-1

A math equations on a graph paper

Description automatically generatedDot products:

**Step b:**

λ1y1 + λ2y2 + λ3y3 = 0

**=>** λ1(1)+ λ2(1) + λ3(1) = 0 (since y1=1, y2=1 and y3= -1)  
=> λ3= λ1+ λ2

Now subsitite x1,x2,x3,y1,y2,y3 in L(λ) = i – ½

A math equations on a graph paper

Description automatically generatedLet’s simplify this further and then substitute one of the lambda values (λ3= λ1+ λ2)

A paper with math equations

Description automatically generated

After this, we can proceed to the next step which is to partially differentiate the ‘L’ we have.

A math equations on a graph paper

Description automatically generated

Now we find the lambda values:

A graph paper with math equations

Description automatically generated

With this, we can also find:

A math problems on a piece of paper

Description automatically generated

**Now let us find the values of ‘w’ and ‘b’**

A math equations on a graph paper

Description automatically generated

A math equations on a graph paper

Description automatically generated

**From all the above, this is what we can gather:**

**‘W’ = weighted vector =(44/95, 4/95)**

**Offset = b= -99/95**

**Here is the separation plane:**A red line with blue dots

Description automatically generated

**SVMs Question 2:**

1. **A general polynomial kernel can be written as K=(aTb + r)^d where a and b are any two points (vectors) in your dataset. Suppose you have a polynomial kernel K specifically with r=0 and d=3. What is your K?**

**Your K (with r=0 and d=3) = ?**

1. **Write your K as a dot product between two vectors. Show all the work.**

Given Equation is **K=(aTb + r)^d.** Let us subsititute the values of r=0 and d=3 in this. We get:

K= (a.b +0 )^3

If we take 2 vectors instead of a and b as a= (a1,a2) and b= (b1,b2), we can do the dot product as:

A graph paper with writing on it

Description automatically generated

Now let us separate a and b values as follows:

A white paper with black text

Description automatically generated

Now let us separate them into coordinates that we can substitute our points in:  
A white tile with black text

Description automatically generated

1. **Choose any 2D point. [For example, (2,3) is a 2D point]. Use your K and show what your 2D point would be in that new kernel space. Show all the steps and work.**

For this example, I chose the points (4,6) which can be solved using the previously obtained equations such as follows:

Thus, for the points 4,6, we obtained them in the new kernal space as:

[64,216,249.41,166.27]

A math equations on a graph paper

Description automatically generated

**Note: Here again, you can find many examples of this type of thing on the SVM slides. Feel free to use them as a guide.**

**Part 3: Doing Data Science**

This part of the test is testing whether you can “be a data scientist”:

– understand a problem

– determine how best to solve it on your own

– and present results and conclusions to a non-technical audience.

There is no one to tell you what to do or how to do this job.

You will make the decisions, determine what to do and how you want to do it, and use your judgement, etc.

**The Problem**

Suppose you work for Amazon and they ask you to (1) determine if a User should get a credit card (yes or no), and (2) determine which products to advertise to the User.

Imagine that you have all of Amazon’s User data – anything you need.

Design/create a small dataset that you can use to address the questions above. Paste it here. Keep it small with 3 – 4 columns and 25 – 30 rows. You decide the column/variable names and what the data would look like. It can be anything you want that also makes sense with respect to the question. You create the dataset you need.

Next, describe, illustrate, and perform (in Python or R) whatever you think you need to do so that you can address the questions asked properly.

The reason why you are not being told what tools/methods/models to use is that on the job – you are the data scientist and you determine what to use, what to do, what the results mean, and how to present it. Please do that here.

Discuss and illustrate your results and conclusions. YOU decide what is needed to do this.

**\*\*Do not include or paste in any code on this document**.

To submit the code used for this question, place your code online (however you want) and include a link to the code here.

Your audience/readers include only **non-technical** people.

I created a dataset with these four columns:

1. Prime membership: This column has data about whether or not a user has an Amazon Prime account
2. Purchase history: This column has details about a user’s purchase history, i.e, the amount they spent on purchases so far in dollars.
3. Engagement: This column describes the user’s interest in buying things based on their wishlist and purchases. This can be in the following categories: Fashion, Lifestyle, Groceries, Digital or Electronics.
4. Location: This column has a customer’s location. It can be inside or outside of US

Based on the above 4 columns, I came up with a new column called credit eligible which has to satisfy 3 conditions:

1. They have to be Amazon Prime members
2. Purchase history has to be greater than $1600
3. They have to be located in the US

Based on these, I got a new column called credit\_eligible. At this point, my data looked like this:

A screenshot of a computer

Description automatically generated

Now that we have this data ready, I went ahead and did some EDA with the data. Here is what I got:

A graph of a distribution of purchase history

Description automatically generated

The above graph shows that purchase history is uniform across one-third of our users. There are about three users in each category who are making the purchases at a higher end of the price category. So in this regard, the data seems balanced.

A blue and orange rectangular bars

Description automatically generated

From the above, we can deduce that half of the population in our dataset does not have a prime membership. Again, there is a balance for this field.

A graph of a bar graph

Description automatically generated with medium confidence

Contrary to the other fields, it seems like the engagement column has a lot of imbalance in data. There seem to be very few users who are making purchases in Fashion and Electronics while all users purchasing digital and grocery goods seems higher.

A graph showing a number of locations

Description automatically generated

This shows that two-thirds of our population is based in the US. Again, there is some imbalance that we can notice here.

A screenshot of a computer screen

Description automatically generated

These statistics suggest that on average, the users are purchasing 1600 dollars. So based on this, I chose the ‘credit\_eligible’ column. Along with that, I also used the conditions that I mentioned above.

Next, I tried to predict the ‘credit\_eligible’ column with the help of all other columns. For this, I had to encode the engagement column. I implemented four models, and the results are as follows:

A graph showing different colored bars

Description automatically generated

Random Forest performed great! We have a good prediction capability when it comes to credit card eligibility. So the imbalances in our data did not have much effect here.

A graph of a graph

Description automatically generated

Here is the graph I made trying to predict the engagement of users. The maximum accuracy I obtained is 0.35 which is much lesser than random chance. To change this, I tried to use various concepts such as hyperparameter tuning, feature engineering, etc to better the model’s performance, but the data inherently has no discernable features or patterns that are capable of predicting the engagement. To further improve our recommendation system, we have to get more data in order to improve diversity and find some patterns.

For a more technical understanding of what I did, please check out the dataset and code here:

<https://github.com/VaishnaviAsuri/Machine-Learning-Exam/blob/main/ML_EXAM.ipynb>