Term Information

Effective Term	Spring 2026

General Information

Course Bulletin Listing/Subject Area	Mathematics
Fiscal Unit/Academic Org	Mathematics - D0671
College/Academic Group	Arts and Sciences
Level/Career	Graduate, Undergraduate
Course Number/Catalog	5638
Course Title	Topics in Risk Modeling II
Transcript Abbreviation	Top Risk Mod 2
Course Description	Risk Modeling is an important area of Actuarial Science, which is based on statistical or machine learning and the underlying mathematics. Math 5637: Topics in Risk Modeling I offers an introduction to these topics, and Math 5638, as the second course in this sequence, covers more advanced machine learning models with a focus on mathematics and theories.
Semester Credit Hours/Units	Fixed: 3

Offering Information

Length Of Course	14 Week, 12 Week, 8 Week, 6 Week
Flexibly Scheduled Course	Never
Does any section of this course have a distance education component?	No
Grading Basis	Letter Grade
Repeatable	No
Course Components	Lecture
Grade Roster Component	Lecture
Credit Available by Exam	No
Admission Condition Course	No
Off Campus	Never
Campus of Offering	Columbus

Prerequisites and Exclusions

Prerequisites/Corequisites	Prereq: 2568 (or equivalent), 5637, 4530 or STAT 4201, and STAT 4202; or department permission.
Exclusions	
Electronically Enforced	Yes

Cross-Listings

Cross-Listings

Subject/CIP Code

Subject/CIP Code	27.0101
Subsidy Level	Doctoral Course
Intended Rank	Junior, Senior, Masters, Doctoral

Requirement/Elective Designation

The course is an elective (for this or other units) or is a service course for other units

Course Details

Course goals or learning objectives/outcomes	• Understand the mathematical concepts and results on which statistical or machine learning is based. • Use modern machine learning methods for modeling cross-sectional and sequential data • Identify the applications of the introduced models in solving real-world finance problems • Implementing basic machine learning models in Python for simple tasks
Content Topic List	• Introduction to ML with cross-sectional data • Basics of Python programming • Probabilistic modeling, Bayesian regression, and Gaussian processes • Feedforward Neural Networks • Sequential modeling and classic time series models • Fitting time series models and making predictions • Probabilistic sequential modeling • Recurrent Neural Networks • Gated Recurrent Units • Convolutional Neural Networks and Autoencoders • Reinforcement Learning Part 1
Sought Concurrence	No

Attachments

- 5638 syllabus.docx: Syllabus

  (Syllabus. Owner: Husen,William J)
- Curriculum_map_actsci_03222025.docx: Curriculum map Act Sci

  (Other Supporting Documentation. Owner: Husen,William J)
- Curriculum_map_math_03222025.docx: Curriculum map Math

  (Other Supporting Documentation. Owner: Husen,William J)
- 5638-Homework-Labs.pdf: Homeworks and Labs

  (Other Supporting Documentation. Owner: Husen,William J)

Comments

- Prerequisites in curriculum changed to match syllabus (STAT 4202 has STAT 4201 or MATH 4530 as a prerequisite, but it better that it is not hidden). Additionally, HMWKs and Labs are attached to address questions about assignments *(by Husen,William J on 09/22/2025 08:22 AM)*
- Please see Subcommittee feedback email sent 05/06/2025. *(by Hilty,Michael on 05/06/2025 09:25 AM)*

Workflow Information

Status	User(s)	Date/Time	Step
Submitted	Husen,William J	03/24/2025 08:50 AM	Submitted for Approval
Approved	Husen,William J	03/24/2025 08:51 AM	Unit Approval
Approved	Vankeerbergen,Bernadette Chantal	04/02/2025 09:30 AM	College Approval
Revision Requested	Hilty,Michael	05/06/2025 09:25 AM	ASCCAO Approval
Submitted	Husen,William J	09/22/2025 08:22 AM	Submitted for Approval
Approved	Husen,William J	09/22/2025 08:22 AM	Unit Approval
Approved	Vankeerbergen,Bernadette Chantal	09/22/2025 08:25 AM	College Approval
Pending Approval	Jenkins,Mary Ellen Bigler Neff,Jennifer Vankeerbergen,Bernadette Chantal Steele,Rachel Lea	09/22/2025 08:25 AM	ASCCAO Approval

Math 5638: Topics in Risk Modeling II

DESCRIPTION

Risk Modeling is an important area of Actuarial Science, which is based on statistical or machine learning and the underlying mathematics. Math 5637: Topics in Risk Modeling I offers an introduction to these topics, and Math 5638, as the second course in this sequence, covers more advanced machine learning models with a focus on mathematics and theories. This course also briefly demonstrates the implementations of these models in Python and their applications in Actuarial Science and Finance.

COURSE OBJECTIVES/LEARNING OUTCOMES

Upon successful completion of the course, students will be able to

- Understand the mathematical concepts and results on which statistical or machine learning is based.
- Use modern machine learning methods for modeling cross-sectional and sequential data
- Identify the applications of the introduced models in solving real-world finance problems
- Implementing basic machine learning models in Python for simple tasks

CLASS FORMAT

Lecture – 3 hours per week

PREREQUISITE

Linear algebra (Math 2568 or equivalent), probability (Math 4530 or Stat 4201 or equivalent), statistics (Stat 4202 or equivalent), and Math 5637; or by department permission.

TEXTS

Class notes will be distributed. The following are recommended textbooks.

- Machine Learning in Finance: From Theory to Practice by Matthew F. Dixon, Igor Halperin, Paul Bilokon (Book PDF available via OSU Library)
- Financial Data Analytics with Machine Learning, Optimization and Statistics by Sam Chen, Ka Chun Cheung, Phillip Yam, Kaiser Fan

HOMEWORK AND EXAMS

There will be

- Five homework assignments
- Three programming assignments (Labs)
- Two midterm exams
- Final exam

EXPECTED WORKLOAD

students will be expected to be working on homework for an approximate total of 6 hours per week.

GRADE

The course grade will be based on

- Homework, 20%
- Labs, 20%
- Two midterm exams, 40%
- Final exam, 20%

Course grade will be determined by the total percentage obtained, roughly as 90–100 for an A, 80–89 for a B, 70–79 for a C, and 60–69 for a D.

SCHEDULE

A tentative weekly schedule is attached. This schedule and the material covered may be changed without notice. It is the students' responsibility to keep track of these changes. Changes may be announced in class verbally, through Carmen, or through email.

ACADEMIC MISCONDUCT

It is the responsibility of the Committee on Academic Misconduct to investigate or establish procedures for the investigation of all reported cases of student academic misconduct. The term "academic misconduct" includes all forms of student academic misconduct wherever committed; illustrated by, but not limited to, cases of plagiarism and dishonest practices in connection with examinations. Instructors shall report all instances of alleged academic misconduct to the committee (Faculty Rule 3335-5-48.7). For additional information, see the Code of Student Conduct at http://studentaffairs.osu.edu/csc/ .

DISABILITY SERVICES STATEMENT

The university strives to maintain a healthy and accessible environment to support student learning in and out of the classroom. If you anticipate or experience academic barriers based on your disability (including mental health, chronic, or temporary medical conditions), please let me know immediately so that we can privately discuss options. To establish reasonable accommodations, I may request that you register with Student Life Disability Services. After registration, make arrangements with me as soon as possible to discuss your accommodations so that they may be implemented in a timely fashion.

If you are ill and need to miss class, including if you are staying home and away from others while experiencing symptoms of a viral infection or fever, please let me know immediately. In cases where illness interacts with an underlying medical condition, please consult with Student Life Disability Services to request reasonable accommodations. You can connect with them at slds@osu.edu; 614-292-3307; or slds.osu.edu.

RELIGIOUS ACCOMMODATION

Ohio State has had a longstanding practice of making reasonable academic accommodations for students' religious beliefs and practices in accordance with applicable law. In 2023, Ohio State updated its practice to align with new state legislation. Under this new provision, students must be in early communication with their instructors regarding any known accommodation requests for religious beliefs and practices, providing notice of specific dates for which they request alternative accommodations within 14 days after the first instructional day of the course. Instructors in turn shall not question the sincerity of a student's religious or spiritual belief system in reviewing such requests and shall keep requests for accommodations confidential.

With sufficient notice, instructors will provide students with reasonable alternative accommodations with regard to examinations and other academic requirements with respect to students' sincerely held religious beliefs and practices by allowing up to three absences each semester for the student to attend or participate in religious activities. Examples of religious accommodations can include, but are not limited to, rescheduling an exam, altering the time of a student's presentation, allowing make-up assignments to substitute for missed class work, or flexibility in due dates or research responsibilities. If concerns arise about a requested accommodation, instructors are to consult their tenure initiating unit head for assistance.

A student's request for time off shall be provided if the student's sincerely held religious belief or practice severely affects the student's ability to take an exam or meet an academic requirement and the student has notified their instructor, in writing during the first 14 days after the course begins, of the date of each absence. Although students are required to provide notice within the first 14 days after a course begins, instructors are strongly encouraged to work with the student to provide a reasonable accommodation if a request is made outside the notice period. A student may not be penalized for an absence approved under this policy.

If students have questions or disputes related to academic accommodations, they should contact their course instructor, and then their department or college office. For questions or to report discrimination or harassment based on religion, individuals should contact the Office of Institutional Equity. (Policy: Religious Holidays, Holy Days and Observances)

Class Schedule (Tentative):

Week 1	Introduction to ML with cross-sectional data
Week 2	Basics of Python programming
Week 3	Probabilistic modeling, Bayesian regression, and Gaussian processes
Week 4	Feedforward Neural Networks Part 1
Week 5	Feedforward Neural Networks Part 2
Week 6	Feedforward Neural Networks Part 3
Week 7	Sequential modeling and classic time series models
Week 8	Fitting time series models and making predictions
Week 9	Probabilistic sequential modeling
Week 10	Recurrent Neural Networks
Week 11	Gated Recurrent Units
Week 12	Convolutional Neural Networks and Autoencoders
Week 13	Reinforcement Learning Part 1
Week 14	Reinforcement Learning Part 2

Homework 1

Math 5638

1. **Exercise 1.1: Market Game**

   Suppose that two players enter into a market game. The rules of the game are as follows: Player 1 is the market maker, and Player 2 is the market taker. In each round, Player 1 is provided with information $\mathbf{x}$, and must choose and declare a value $\alpha \in \{0, 1\}$ that determines how much it will pay out if a binary event G occurs in the round. $G \sim \text{Bernoulli}(p)$, where $p = g(\mathbf{x}|\theta)$ for some unknown parameter θ.

   Player 2 then enters the game with a \$1 payment and chooses one of the following payoffs:

$$V_1(G, p) = \begin{cases} \frac{1}{\alpha} & \text{with probability } p \\ 0 & \text{with probability } (1-p) \end{cases}$$

   or

$$V_2(G, p) = \begin{cases} 0 & \text{with probability } p \\ \frac{1}{1-\alpha} & \text{with probability } (1-p) \end{cases}$$

   Questions:

   (a) Given that α is known to Player 2, state the strategy that will give Player 2 an expected payoff, over multiple games, of \$1 without knowing p.

   (b) Suppose now that p is known to both players. In a given round, what is the optimal choice of α for Player 1?

2. **Exercise 1.4: Likelihood Estimation**

   When the data is i.i.d., the negative of log-likelihood function (the "error function") for a binary classifier is the *cross-entropy*

$$E(\theta) = -\sum_{i=1}^{n} G_i \ln\left(g_1(x_i \mid \theta)\right) + (1 - G_i) \ln\left(g_0(x_i \mid \theta)\right).$$

   Suppose now that there is a probability π_i that the class label on a training data point x_i has been correctly set. Write down the error function corresponding to the negative log-likelihood. Verify that the error function in the above equation is obtained when $\pi_i = 1$. Note that this error function renders the model robust to incorrectly labeled data, in contrast to the usual least squares error function.

3. **Exercise 1.5: Optimal Action**

   Derive Eq. **1.17** by setting the derivative of Eq. **1.16** with respect to the time-t action u_t to zero. Note that Eq. **1.17** gives a non-parametric expression for the optimal action u_t in terms of a ratio of two conditional expectations. To be useful in practice, the approach might need some further modification as you will use in the next exercise.

4. **Exercise 1.6: Basis Functions**

   Instead of non-parametric specifications of an optimal action in Eq. 1.17, we can develop a parametric model of optimal action. To this end, assume we have a set of basis functions $\psi_k(S)$ with $k = 1, \ldots, K$. Here K is the total number of basis functions—the same as the dimension of your model space.

   We now define the optimal action $u_t = u_t(S_t)$ in terms of coefficients $\theta_k(t)$ of expansion over basis functions Ψ_k (for example, we could use spline basis functions, Fourier bases, etc.):

$$u_t = u_t(S_t) = \sum_{k=1}^{K} \theta_k(t) \Psi_k(S_t).$$

   Compute the optimal coefficients $\theta_k(t)$ by substituting the above equation for u_t into Eq. 1.16 and maximizing it with respect to a set of weights $\theta_k(t)$ for a t-th time step.

Homework 2

Math 5638

1. **Exercise 2.2: FX and Equity**

   A currency strategist has estimated that JPY will strengthen against USD with probability 60% if S&P 500 continues to rise. JPY will strengthen against USD with probability 95% if S&P 500 falls or stays flat. We are in an upward trending market at the moment, and we believe that the probability that S&P 500 will rise is 70%. We then learn that JPY has actually strengthened against USD. Taking this new information into account, what is the probability that S&P 500 will rise?

   Hint: Recall Bayes' rule:

   $$P(A \mid B) = \frac{P(B \mid A)}{P(B)} P(A).$$

2. **Exercise 2.4: Bayesian Inference in Trading**

   Suppose that you observe the following daily sequence of directional changes in the JPY/USD exchange rate (U (up), D (down or stays flat)):

   U, D, U, U, D

   and the corresponding daily sequence of S&P 500 returns is

   $$-0.05,\ 0.01,\ -0.01,\ -0.02,\ 0.03$$

   You propose the following probability model to explain the behavior of JPY against USD given the directional changes in S&P 500 returns: Let G denote a Bernoulli random variable, where $G = 1$ corresponds to JPY strengthening against the dollar and r are the S&P 500 daily returns. All observations of G are conditionally independent (but *not* identical) so that the likelihood is

   $$p(G \mid r, \theta) = \prod_{i=1}^{n} p(G = G_i \mid r = r_i, \theta)$$

   where

   $$p(G_i = 1 \mid r = r_i, \theta) = \begin{cases} \theta_u, & r_i > 0 \\ \theta_d, & r_i \leq 0. \end{cases}$$

Compute the full expression for the likelihood that the data was generated by this model.

3. **Exercise 2.7: Logistic Regression Is Naive Bayes**

   Suppose that G and $X \in \{0,1\}^p$ are Bernoulli random variables and the X_is are mutually independent given G—that is,

$$P[X \mid G] = \prod_{i=1}^{p} P[X_i \mid G].$$

   Given a naive Bayes' classifier $P[G \mid X]$, show that the following logistic regression model produces equivalent output if the weights are

$$w_0 = \log \frac{P[G]}{P[G^c]} + \sum_{i=1}^{p} \log \frac{P[X_i = 0 \mid G]}{P[X_i = 0 \mid G^c]}.$$

$$w_i = \log \frac{P[X_i = 1 \mid G]}{P[X_i = 1 \mid G^c]} - \log \frac{P[X_i = 0 \mid G]}{P[X_i = 0 \mid G^c]}, \quad i = 1, \dots, p.$$

4. **Exercise 3.2: Normal Conjugate Distributions**

   Suppose that the prior is

$$p(\theta) = \phi(\theta; \mu_0, \sigma_0^2)$$

   and the likelihood is given by

$$p(x_{1:n} \mid \theta) = \prod_{i=1}^{n} \phi(x_i; \theta, \sigma^2),$$

   where σ^2 is assumed to be known. Show that the posterior is also normal,

$$p(\theta \mid x_{1:n}) = \phi(\theta; \mu_{\text{post}}, \sigma_{\text{post}}^2),$$

   where

$$\mu_{\text{post}} = \frac{\frac{\bar{x}}{\sigma^2/n} + \frac{\mu_0}{\sigma_0^2}}{\frac{1}{\sigma^2/n} + \frac{1}{\sigma_0^2}}, \qquad \sigma_{\text{post}}^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}},$$

   and

$$\bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Homework 3

Math 5638

1. **Exercise 4.2**

   Show that substituting the derivative of the softmax function w.r.t. w_{ij} into Eq. 4.52 gives for the special case when the output is $Y_k = 1$, $k = i$, and $Y_k = 0$, $\forall k \neq i$:

$$\nabla_{ij}\mathcal{L}(W, b) := [\nabla_W \mathcal{L}(W, b)]_{i,j} = \begin{cases} (\sigma_i - 1)X_j, & Y_i = 1, \\ 0, & Y_k = 0, \forall k \neq i. \end{cases}$$

   (for this question, we consider a multiclass classification problem and a feedforward neural network with only one hidden layer; Read the part above Section 5.1.1 to get familiar with the notations. Also, refer to our class notes for the derivation details.)

2. **Exercise 4.3**

   Consider feedforward neural networks constructed using the following two types of activation functions:

   - Identity

$$\mathrm{Id}(x) := x$$

   - Step function (a.k.a. Heaviside function)

$$H(x) := \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

   (a) Consider a feedforward neural network with one input $x \in \mathbb{R}$, a single hidden layer with K units having step function activations $H(x)$, and a single output with identity (a.k.a. linear) activation $\mathrm{Id}(x)$. The output can be written as

$$\hat{f}(x) = \mathrm{Id}\left(b^{(2)} + \sum_{k=1}^{K} w_k^{(2)} H\left(b_k^{(1)} + w_k^{(1)} x\right)\right).$$

   Construct neural networks using these activation functions.

   i. Consider the step function

$$u(x; a) := y\, H(x - a) = \begin{cases} y, & \text{if } x \geq a, \\ 0, & \text{otherwise.} \end{cases}$$

Construct a neural network with one input x and one hidden layer, whose response is $u(x; a)$. Draw the structure of the neural network, specify the activation function for each unit (either Id or H), and specify the values for all weights (in terms of a and y).

ii. Now consider the indicator function

$$1_{[a,b)}(x) = \begin{cases} 1, & \text{if } x \in [a, b), \\ 0, & \text{otherwise.} \end{cases}$$

Construct a neural network with one input x and one hidden layer, whose response is $1_{[a,b)}(x)$, for given real values a and b. Draw the structure of the neural network, specify the activation function for each unit (either Id or H), and specify the values for all weights (in terms of a, b, and y).

(This problem is not clearly written. I uploaded the solution to 4.3(a) as a demonstration of what you are asked to do. Also refer to the notes for Class 9 for inspiration.)

3. **Exercise 4.4**

A neural network with a single hidden layer can provide an arbitrarily close approximation to any 1-dimensional bounded smooth function. This question will guide you through the proof. Let $f(x)$ be any function whose domain is $[C, D]$, for real values $C < D$. Suppose that the function is Lipschitz continuous, that is,

$$\forall x, x' \in [C, D],\ |f(x') - f(x)| \leq L|x' - x|,$$

for some constant $L \geq 0$. Use the building blocks constructed in the previous part to construct a neural network with one hidden layer that approximates this function within $\epsilon > 0$, that is,

$$\forall x \in [C, D],\ |f(x) - \hat{f}(x)| \leq \epsilon,$$

where $\hat{f}(x)$ is the output of your neural network given input x. Your network should use only the identity or the Heaviside activation functions. You need to specify:

- the number K of hidden units,
- the activation function for each unit,
- and a formula for calculating each weight

$$w_0,\ w_k,\ w_0^{(k)},\ w_1^{(k)},$$

for each $k \in \{1, \ldots, K\}$.

These weights may be specified in terms of

$$C, D, L, \text{ and } \epsilon,$$

as well as the values of $f(x)$ evaluated at a finite number of x values of your choosing (you need to explicitly specify which x values you use). You do not need to explicitly write the $\hat{f}(x)$ function. Why does your network attain the given accuracy ϵ? (This problem depends on 4.3; Also, refer to this webpage presented in class for ideas: http://neuralnetworksanddeeplearning.com/chap4.html)

4. **Exercise 4.5**

Consider a shallow neural network regression model with n tanh-activated units in the hidden layer and d outputs. The hidden–outer weight matrix $W_{ij}^{(2)} = \frac{1}{n}$ and the input–hidden weight matrix $W^{(1)} = 1$. The biases are zero. If the features $X_1, \ldots, X_p$ are i.i.d. Gaussian random variables with mean $\mu = 0$ and variance σ^2, show that:

(a) $\hat{Y} \in [-1, 1]$.

(b) $\hat{Y}$ is independent of the number of hidden units, $n \geq 1$.

(c) The expectation $E[\hat{Y}] = 0$ and the variance $V[\hat{Y}] \leq 1$.

Homework 4

Math 5638

1. **Exercise 6.1**

   Calculate the mean, variance, and autocorrelation function (ACF) of the following zero-mean AR(1) process:
   $$y_t = \phi_1 y_{t-1} + \epsilon_t,$$
   where $\phi_1 = 0.5$. Determine whether the process is stationary by computing the root of the characteristic equation $\Phi(z) = 0$.

2. **Exercise 6.2**

   You have estimated the following ARMA(1,1) model for some time series data:
   $$y_t = 0.036 + 0.69 y_{t-1} + 0.42 u_{t-1} + u_t,$$
   where you are given the data at time $t-1$: $y_{t-1} = 3.4$ and $\hat{u}_{t-1} = -1.3$. Obtain the forecasts for the series y for times $t, t+1, t+2$ using the estimated ARMA model.

   If the actual values for the series are $-0.032, 0.961, 0.203$ for $t, t+1, t+2$, calculate the out-of-sample Mean Squared Error (MSE) and Mean Absolute Error (MAE).

3. **Exercise 6.3**

   Derive the mean, variance, and autocorrelation function (ACF) of a zero-mean MA(1) process.

4. **Exercise 6.6**

   Suppose that, for the sequence of random variables $\{y_t\}_{t=0}^{\infty}$, the following model holds:
   $$y_t = \mu + \phi y_{t-1} + \epsilon_t, \quad |\phi| \leq 1, \quad \epsilon_t \sim \text{i.i.d.}(0, \sigma^2).$$
   Derive the conditional expectation $E[y_t \mid y_0]$ and the conditional variance $\text{Var}[y_t \mid y_0]$.

Homework 5

Math 5638

1. Based on Example 7.1 in the textbook, and suppose we do not know the hidden sequence, find the marginal probability of the observed sequence. That is, find

$$\mathbb{P}(\{y_1, y_2, y_3\} = \{-1, 1, 1\} \mid A, B, \pi)$$

without using the information given in the example that the hidden state sequence is $\{1, 0, 0\}$. What algorithm should we use?

2. Again, based on Example 7.1 and suppose the hidden sequence is unknown. Find the state probability

$$\mathbb{P}(S_1 = 1 \mid A, B, \pi, y_1, y_2, y_3).$$

What algorithm do we use?

3. Second question of Exercise 8.2. Show that a linear RNN(p) model with bias terms in both the output layer and the hidden layer can be written in the form

$$\hat{y}_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i}$$

and state the form of coefficients ϕ_i.

MATH 5194 AU 24 Lab 1

Instructions

Please do the following for this lab. Don't forget to upload your .ipynb file to CarmenCanvas Assignments

1. Simulate data from a binary logit model. More details below.
2. Save the data as /binarylogitdata.csv
3. Estimate OLS model
4. Estimate Logit model with L2 penalty

Simulate data from a binary logit model

First thing we want to do is import packages that will be useful for simulating and estimating our model:

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import scipy.optimize as opt
```

Next, let's generate our data. There are $N = 1000$ observations and for each observation, we have (y_i, x_i) where

$$y_i = \mathbf{1}_{\{x_i\beta - \epsilon_i > 0\}},$$

- x_i are realizations of independent and identically distributed random variables (iid rvs) drawn from a standard lognormal distribution and
- ϵ_i are realizations of iid rvs drawn from a logistic distribution with the following CDF,

$$F(x_i) = \frac{\exp(x_i\beta)}{1 + \exp(x_i\beta)}.$$

Finally, set $\beta = 0.3$.

```
np.random.seed(1234)
## Parameter settings

## Generate data

## Data prep and save

# output first 10 lines of data (d,x)

# save dataset
```

Estimate OLS

- find the value of the least-squares estimator
  - $\beta = (X'X)^{-1}X'Y$
  - note that the model includes an intercept/bias term, therefore the X matrix should have a leading column of 1
- calculate the fitted values and plot them along with the true values

```

```

Estimate logit model

- estimate the parameters using logistic regression with a L2 penalty term
- the hyperparameter should be tuned using k-fold cross validation (e.g., $k = 5$)

- calculate and report the cross-entropy of your model

```
Start coding or generate with AI.
```

MATH 5194 AU 24 Lab 2

Instructions

Please do the following for this lab. Don't forget to upload your .ipynb file to CarmenCanvas Assignments

- calculate the prices of a bull call spread under the Black-Scholes framework for different spot prices (current stock prices)
- create samples for building FNN using `pytorch`
- identify the contraints that we need to consider when using FNN to approximate the relationship between the bull spread and the spot price
- build the feedforward neural network model and plot the results

Bull Call Spread

We can create a portfolio called the bull spread using call options.

Specifically, to create this portfolio, we fix the underlying asset and the expiration date, buy one call option with the strike price being K_1 and simultaneously sell/write another call option with strike price $K_2 > K_1$. The price of this portfolio is the price of the option we buy minus the price of the option we sell.

For this lab, we consider option prices under the Black-Scholes framework.

Load libraries

```
# DO NOT modify this block
# load libraries
from google.colab import drive
drive.mount('/content/drive')
import sys
sys.path.append('/content/drive/MyDrive/Colab Notebooks')
import matplotlib.pyplot as plt
import numpy as np
from BlackScholes import bsformula # this is from the BlackScholes.py script I uploaded to Carmen

# import pytorch
import torch
import torch.nn as nn
import torch.optim as optim
```

```
Mounted at /content/drive
```

Set up option parameters

The strike prices of the two call options are as follows,

- $K_1 = 130$
- $K_2 = 140$

```
# DO NOT modify this block
# BS parameters
r = 0.0002           # risk-free rate
KC1 = 130            # long call strike
KC2 = 140            # short call strike
sigma = 0.4          # implied volatility
T = 2.0              # Time to maturity
lb = 0.001           # lower bound on domain
ub = 600             # upper bound on domain
```

Calculate call option prices and the spread price

Modify the following code to make

- `call1` calculate the price of the call option with strike price K_1,

- `call2` calculate the price of the call option with strike price K_2, and
- `bullSpread` calculate the price of the bull spread.

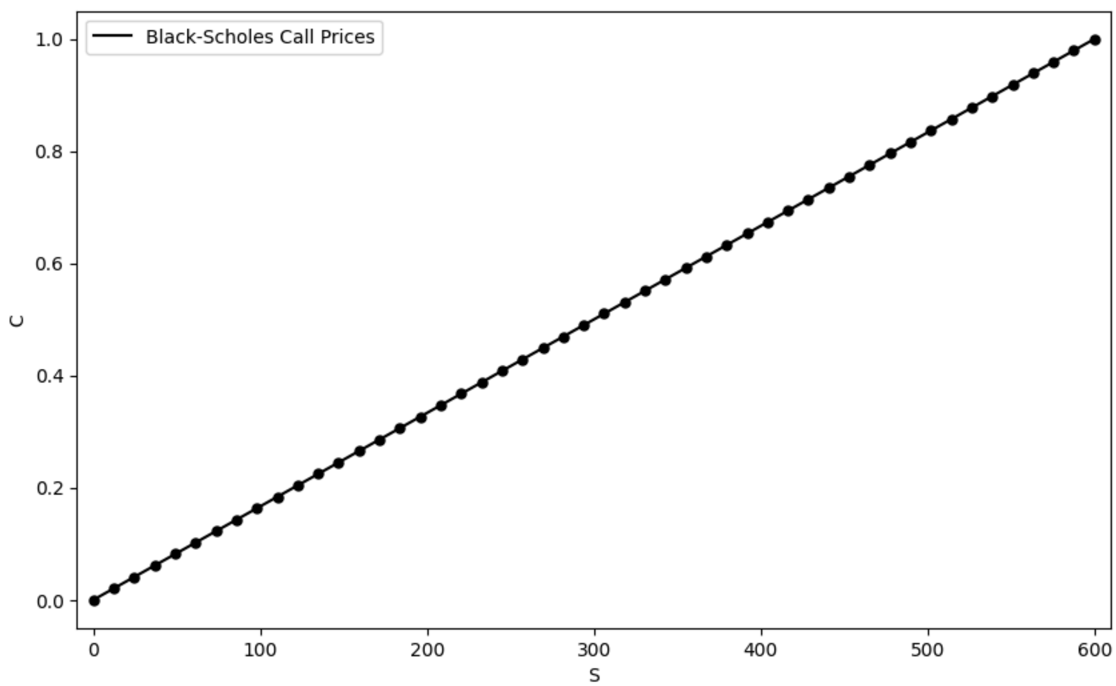
```
# modify the following functions
call1 = lambda x: x
call2 = lambda x: x
bullSpread = lambda x: x
```

```
# DO NOT modify this block
# build the sample
training_number = 50  # Number of training samples
testing_number = 100 # Number of testing samples

# draw evenly-spaced values from the input space, stock prices
train_x_1 = np.array(np.linspace(0, 1, training_number), dtype='float32').reshape(training_number, 1)
train_y_1 = [bullSpread(S) for S in train_x_1] # give a list of bull spread prices
test_x_1 = np.array(np.linspace(0, 1, testing_number), dtype='float32').reshape(testing_number, 1)
test_y_1 = [bullSpread(S) for S in test_x_1]
```

```
# DO NOT modify this block
# plot the sample
plt.figure(figsize = (10, 6), facecolor='white', edgecolor='black')
plt.plot(lb+(ub-lb)*test_x_1.flatten(), test_y_1, color = 'black', label = 'Black-Scholes Call Prices')
plt.scatter(lb+(ub-lb)*train_x_1, train_y_1, color = 'black', marker = '.', s = 100)
plt.legend(loc = 'best', prop={'size':10})
plt.xlim([lb-10, ub+10])
plt.xlabel('S')
plt.ylabel('C')
```

```
Text(0, 0.5, 'C')
```



Constraints

If you do the previous part correctly, you should find that the price of the spread is bounded.

- State the bounds of the price. Briefly explain why the price bounded by those values. If you have no clue, ask your classmates who have learned about option pricing for help.
  - **Write your answer here:**
- If we use a one-hidden-layer FNN to approximate the bull spread price as a function of the spot price, what properties should the weight matrices, bias vectors, and activation functions have?
  - $W^{(1)}$:

- $b^{(1)}$:
- $\sigma^{(1)}$:
- $W^{(2)}$:
- $b^{(2)}$:
- $\sigma^{(2)}$:

Sample

Convert the samples to tensors so that they can be used by `pytorch`

```
## convert the samples to tensors
```

Build the feedforward neural network

```
# specify the structure of the neural network in this block
```

```
# define any necessary functions to process weights and biases in this block
```

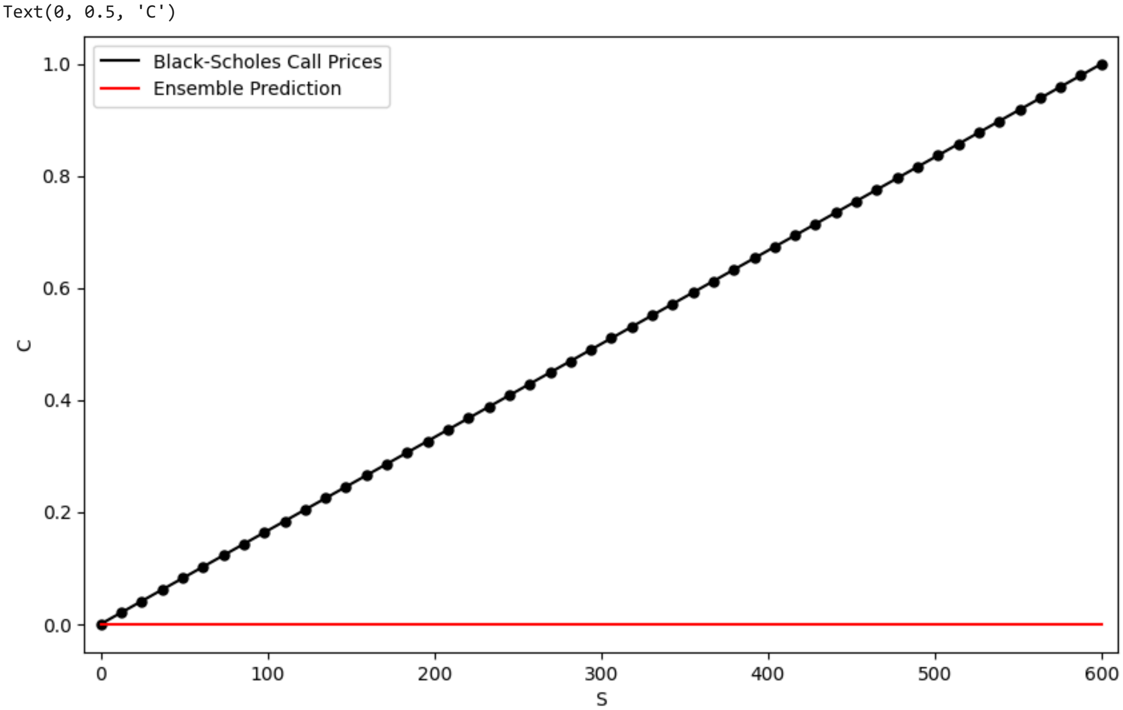
```
# specify the loss function and optimizer in this block
```

```
# use minibatch stochastic gradient descent to train the model in this block
```

```
# modify the code in this block and let y_pred be the predictions made on the test set
y_pred = np.zeros(len(test_x_1))
```

```
# DO NOT modify this block
# plot
plt.figure(figsize = (10, 6), facecolor='white', edgecolor='black')
plt.plot(lb+(ub-lb)*test_x_1.flatten(), test_y_1, color = 'black', label = 'Black-Scholes Call Prices')
plt.plot(lb+(ub-lb)*test_x_1.flatten(), y_pred, color = 'red', label = 'Ensemble Prediction')
plt.scatter(lb+(ub-lb)*train_x_1, train_y_1, color = 'black', marker = '.', s = 100)
plt.legend(loc = 'best', prop={'size':10})
plt.xlim([lb-10, ub+10])
plt.xlabel('S')
plt.ylabel('C')
```

```
Text(0, 0.5, 'C')
```



MATH 5194 AU 24 Lab 3

Instructions

Follow the instructions in each block to complete the code. Upload your .ipynb file to CarmenCanvas Assignments

```
# DO NOT modify this block
# load libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from google.colab import drive
drive.mount('/content/drive')
import sys
sys.path.append('/content/drive/MyDrive/Colab Notebooks')
import torch
import torch.nn as nn
import torch.optim as optim
import torch.utils.data as data
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Modify this block to read the data file sp500_close.csv
df = pd.DataFrame()
df = df.values.reshape(-1)
```

Create Training/Test Samples for Cross Validation

The time series data has 1205 observations in the sequence.

- Create two sets of training and test samples following the approach described in class
- Each set has 805 observations in the training sample and 200 observations in the test sample

```
# Modify this block
# Make the function below generate a list of training samples and a list of test samples
def createTrainTest(dataset, train_size, test_size):
  train_df, test_df = [], []
  # add your code here
  return train_df, test_df
```

```
# DO NOT Modify this block
# Use the function above to create the training and test samples
train_size = 805
test_size = 200
train_list, test_list = createTrainTest(df, train_size, test_size)
```

Create Features and Responses

For each set of training and test samples, using every 40 consecutive values as features (lookback) and the corresponding responses are the sequence of observations in the 40-day window shifted forward by one observations. For example, in the first training sample

- the first sequence of features starts with index 0 and ends with index 39
  - correspondingly, the sequence of responses starts with index 1 and ends with index 40.
- the second sequence of features starts with index 1 and ends with index 40
  - correspondingly, the sequence of responses starts with index 2 and ends with index 41.
- etc.

```
# Modify this block
# Make the function below generate a tensor of features X and a tensor of responses y
def createXy(dataset, lookback):
    X, y = [], []
    # add your code here
    return torch.tensor(X, dtype=torch.float), torch.tensor(y, dtype=torch.float)
```

```
# DO NOT modify this block
# Work with the first pair of training and test set for now
train, test = train_list[0], test_list[0]
```

Scaling (Please read)

- large values in time series can make it difficult to choose a proper learning rate for gradient descent
- scaling the data to smaller values in a smaller range can make the fitting of the model easier
- two common ways of scaling
  - standardizing: shift the data by its mean (centering) and scale it by its standard deviation; This makes the data have zero mean and unit variance
$$x' = \frac{x - \bar{x}}{\sigma_x}$$
  - min/max scaling: shift the data by its minimum and scale it by the difference between the maximum and minimum values; This rescales the data to the range $[0, 1]$
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$
- Only the values in the training set should be used to determining the scaling parameters (e.g., mean and standard deviation, or min and max)
  - To avoid *look-ahead bias* - using the information that is supposedly unknown for scaling.
  - The test set should be scaled accordingly
- Example (min/max):
  - training set: $\{1, 2, 3\}$; test set: $\{4, 5, 6\}$
  - minimum of training set: 1; maximum of training set: 3
  - scaled training set: $\left\{\frac{1-1}{3-1}, \frac{2-1}{3-1}, \frac{3-1}{3-1}\right\} = \{0, 0.5, 1\}$
  - scaled test set: $\left\{\frac{4-1}{3-1}, \frac{5-1}{3-1}, \frac{6-1}{3-1}\right\} = \{1.5, 2, 2.5\}$
- The example in class 25 used min/max scaling. For this lab, please use standardization method to scale the data. You can use `StandardScaler` from scikit learn.

```
# Modify this block
# Standardizing the training and test sets created above.
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
train = []
test = []
```

```
# DO NOT modify this block
# Generate features with a lookback period of 40 and the corresponding responses
lookback = 40
X_train, y_train = createXy(train, lookback)
X_test, y_test = createXy(test, lookback)
```

Fit GRU Model to Data

```
# Modify this block
# Implement a GRU model

class GRU(nn.Module):
    def __init__(self):
      return None
```

```
# DO NOT modify this block
model = GRU()
loader = data.DataLoader(data.TensorDataset(X_train, y_train), batch_size=20)
```

```
# Modify this block

# specify an SGD optimizer
optimizer = None
# specify an MSR loss function
loss_fn = None
# specify the number of epochs
n_epochs = 0


for epoch in range(n_epochs):
    model.train()
    for X_batch, y_batch in loader:
        # specify the forward pass
        y_pred = model(X_batch)
        # evaluate the loss
        loss = loss_fn(y_pred, y_batch)
        # backward propagation
        optimizer.zero_grad()
        loss.backward()
        # update the parameters
        optimizer.step()
    # DO NOT modify the code below
    # Validation
    if epoch % 100 != 0:
        continue
    model.eval()
    with torch.no_grad():
        y_pred = model(X_train)
        train_mse = loss_fn(y_pred, y_train)
        y_pred = model(X_test)
        test_mse = loss_fn(y_pred, y_test)
    print("Epoch %d: train MSE %.4f, test MSE %.4f" % (epoch, train_mse, test_mse))
```

```
# DO NOT modify this block
# Plotting
ts = np.append(train, test)
with torch.no_grad():
    train_plot = np.ones_like(ts) * np.nan
    y_pred = model(X_train)[:, -1]
    train_plot[lookback:train_size] = y_pred
    test_plot = np.ones_like(ts) * np.nan
    test_plot[train_size+lookback:len(ts)] = model(X_test)[:, -1]
# plot
plt.plot(ts, c='grey')
plt.plot(train_plot, c='b')
plt.plot(test_plot, c='orange')
plt.show()
```

Hyperparameter tuning

Previously, we used a lookback period of 40 days. The lookback period can be treated as a hyperparameter.

Suppose we need to decide whether a 40-day lookback period is better or a 20-day lookback period is better.

Briefly describe how we should make this decision. Your answer should mention cross validation and the second pair of training and test sets we created above.

Your answer here:

Math - BS/BA Curriculum Map					
Goal 1	**Learn conceptual frameworks needed to study higher mathematics, including an introduction to mathematical reasoning and an understanding of how to read and write proofs.**				
Goal 2	**Aquire basic mastery of core areas of mathematics including calculus, analysis and algebra.**				
Goal 3	**Develop powerful mathematical problem solving skills.**				
Goal 4	**Learn to communicate mathematical understanding effectively.**				
Goal 5	**Become proficient in chosen tracks within the major.**				
Course	**Goal 1**	**Goal 2**	**Goal 3**	**Goal 4**	**Goal 5**
AcctMIS 2000			Beginning		Intermediate
Biochem 4511					Advanced
Biology 1113			Beginning		Intermediate
Biology 1114			Beginning		Intermediate
Biology 3401					Intermediate
BusFin 3120			Intermediate	Intermediate	Advanced
BusFin 3220			Intermediate	Intermediate	Advanced
Chem 1210			Beginning		Intermediate
Chem 1220			Beginning		Intermediate
Chem 2210					Advanced
Chem 2510					Advanced
Chem 4300					Advanced
Chem 4310					Advanced
CSE 1222			Beginning		Intermediate
CSE 1223			Beginning		Intermediate
CSE 2221			Beginning	Beginning	
CSE 2111			Beginning		Intermediate
Econ 2001.01			Beginning		Intermediate
Econ 2002.01			Beginning		Intermediate
EEOB 3310					Advanced
EEOB 3420					Advanced
EEOB 4520					Advanced
Math 1151	Beginning	Beginning	Beginning		
Math 1152	Beginning	Beginning	Beginning		
Math 1181H	Intermediat e	Intermediate	Beginning		
Math 1295				Intermediate	Beginning
Math 2153	Intermediat e	Intermediate	Beginning		

Math 2182H	Intermediate	Intermediate	Beginning		
Math 2255	Beginning	Intermediate	Intermediate	Beginning	
Math 2568	Beginning	Beginning	Beginning		Beginning
Math 2568H	Intermediate	Beginning	Intermediate	Beginning	Beginning
Math 3345	Advanced	Advanced	Intermediate	Intermediate	Intermediate
Math 3345H	Advanced	Advanced	Intermediate	Intermediate	Intermediate
Math 3350				Intermediate	Beginning
Math 3589			Intermediate	Intermediate	Advanced
Math 3607			Intermediate	Intermediate	Advanced
Math 3618			Intermediate	Advanced	Advanced
Math 4181H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 4182H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 4345	Advanced	Advanced	Advanced	Intermediate	Advanced
Math 4350			Intermediate	Advanced	Advanced
Math 4504	Advanced	Intermediate	Intermediate	Advanced	Advanced
Math 4507	Advanced	Intermediate	Intermediate	Advanced	Advanced
Math 4512	Intermediate		Intermediate	Intermediate	Intermediate
Math 4530	Intermediate	Beginning	Intermediate	Intermediate	Intermediate
Math 4547	Advanced	Advanced	Intermediate	Advanced	Beginning
Math 4548	Advanced	Advanced	Intermediate	Advanced	Beginning
Math 4551	Intermediate	Intermediate	Intermediate	Intermediate	Intermediate
Math 4552	Intermediate	Intermediate	Intermediate	Intermediate	Intermediate
Math 4556			Intermediate	Advanced	Advanced
Math 4557	Intermediate		Intermediate	Intermediate	Intermediate
Math 4570	Intermediate	Intermediate	Advanced	Intermediate	Intermediate
Math 4573	Advanced	Intermediate	Intermediate	Intermediate	Intermediate
Math 4575	Intermediate	Intermediate	Intermediate	Intermediate	Intermediate
Math 4578	Intermediate	Intermediate	Intermediate	Intermediate	Advanced
Math 4580	Advanced	Advanced	Intermediate	Advanced	Beginning
Math 4581	Advanced	Advanced	Intermediate	Advanced	Beginning
Math 5101	Beginning	Advanced	Intermediate		Intermediate
Math 5102	Beginning	Advanced	Intermediate		Intermediate
Math 5421	Beginning	Beginning	Intermediate	Beginning	Advanced
Math 5451	Beginning	Beginning	Intermediate	Beginning	Advanced

Math 5520H	Advanced	Advanced	Advanced	Advanced	Intermediate
Math 5522H	Advanced	Advanced	Advanced	Advanced	Intermediate
Math 5529H	Advanced	Advanced	Advanced	Advanced	Intermediate
Math 5530H	Advanced	Advanced	Advanced	Advanced	Intermediate
Math 5540H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 5540H	Advanced	Advanced	Advanced	Intermediate	Beginning
Math 5571	Advanced	Advanced	Advanced	Intermediate	Intermediate
Math 5576H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 5590H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 5591H	Advanced	Advanced	Advanced	Advanced	Advanced
Math 5632			Intermediate	Advanced	Advanced
Math 5635			Intermediate	Advanced	Advanced
Math 5636			Intermediate	Advanced	Advanced
Math 5637			Intermediate	Advanced	Advanced
Math 5638			**Intermediate**	**Advanced**	**Advanced**
Math 5660					Intermediate
Math 5756			Beginning	Intermediate	Intermediate
Math 5757			Beginning	Intermediate	Intermediate
MolGen 4500					Advanced
MolGen 5601					Advanced
Physics 1250			Beginning		Intermediate
Physics 1251			Beginning		Intermediate
Physics 2300					Advanced
Physics 2301					Advanced
Stat 4201	Intermediate	Beginning	Intermediate	Intermediate	Intermediate
Stat 4202	Intermediate		Intermediate		Intermediate

Actuarial Science BS/BA Curriculum Map					
Goal 1	**To supply a strong general background in mathematics, statistics, and relevant concepts from the insurance industry**				
Goal 2	**To prepare students to take some of the national actuarial examinations administered by the Society of Actuaries and the Casualty Actuarial Society**				
Course	**Goal 1**	**Goal 2**			
Math 1151	Beginning	Beginning			
Math 1152	Beginning	Beginning			
ACCTMIS 2000	Beginning				
Econ 2001.01	Beginning				
Econ 2002.01	Beginning				
CSE 1222	Beginning	Intermediate			
CSE 1223	Beginning	Intermediate			
CSE 2111	Beginning	Intermediate			
Comm 2110	Beginning				
Comm 2131	Beginning				
Comm 2367	Beginning				
BusFin 3120	Intermediate	Beginning			
English 3304	Beginning				
Math 2153	Intermediate	Beginning			
Math 2568	Intermediate	Beginning			
Math 3588	Intermediate	Advanced			
Math 3618	Intermediate	Advanced			
Math 4530	Advanced	Advanced			
Stat 4201	Advanced	Advanced			
Math 5632	Advanced	Advanced			
Stat 4202	Advanced	Advanced			
Math 5571	Advanced	Advanced			
Math 5630	Advanced	Advanced			
Math 5631	Advanced	Advanced			
Math 5633	Advanced	Advanced			
Math 5634	Advanced	Advanced			
Math 5635	Advanced	Advanced			
Math 5636	Advanced	Advanced			
Math 5637	Advanced	Advanced			
Math 5638	**Advanced**	**Advanced**			