Columbia University

MATH GR5260 Spring 2023

Programming for Quant and Computational Finance

Ka Yi Ng

Exam 2 Sample Questions

The purpose of this document is to let you get familiar with the exam style and format. The actual exam may include questions on completely different topics.

GUIDELINES

This is an open book exam. You may use any notes, reference materials, internet, Jupyter Notebook or any Python IDEs during the exam period. However, you only have limited time to complete the exam, so you must use your time wisely.

All answers must be written legibly.
In the Python code, add comments to your code where you think will help grader understand your logic.

Solution submission

- Answer the theory questions on the exam paper provided.
- Submit your python code (and output results) as HTML or pdf or Jupyter notebook files onto Courseworks under 'Exam 2' section, in the same way you uploaded your homework previously. If you have technical issues in submitting your solution, please notify your TA or Prof Ng and email your solution directly to your TA to get a proper timestamp of submission.
- You may name these files whichever way you like.
- Make sure that in each of these files, you have your name and UNI on it.

During the exam, if you suspect of any typos or have questions, notify your TA or Prof Ng.

Prof Ng: kyn@math.columbia.edu
Ying Wu: yw3521@columbia.edu

HONOR CODE AFFIRMATION

I affirm that I will not plagiarize, use unauthorized materials, or give or receive illegitimate help on assignments, papers, or examinations. I will also uphold equity and honesty in the evaluation of my work and the work of others. I do so to sustain a community built around this Code of Honor.
(https://www.college.columbia.edu/honorcode#:~:text=)

First name	Last name	UNI	Signature

QUESTION 1

This question is to investigate and compare the 10-day VaR calculations using a 10-day window and a 1-day window. We will use the Dow Jones Index closing prices from Yahoo Finance as the data source.

Hypothetically, suppose we have held a portfolio of 1 share of DJI for the past 2 years.

The current date is April 21st 2023.

a) Compute the 95% 1-day historical VaR of the portfolio for the past 100 days. Each historical VaR value (say, on date T) shall be computed using 250 days of daily log price returns (ie. returns on T-249, …, T-1, T). Display the result for the current date.

b) Estimate 95% 10-day VaR for the past 100 days from the results in (a). (Use: N-day VaR = 1-day VaR x $\sqrt{N}$) Display the result for the current date.

c) Compute the 95% 10-day historical VaR of the portfolio for the past 100 days. Each historical VaR value shall be computed using 250 days of 10-day log price returns. Display the result for the current date.

d) Plot the results of part (b) and (c) in the same figure with proper labels.

e) Comment on the shape of the plots in (d). Could you give an explanation for the shape of the plots?

QUESTION 2

This question is about comparing different machine learning models in the prediction of option prices.

The attached file, option_data.csv, includes simulated prices of a European call option with different spot prices, strikes, time to expiry (in years) and volatilities.

a) Use option_data.csv as the training set to train the following models to predict option prices. Use grid search with 5-fold cross validation to find the optimal hyperparameters when requested below.

   i. SVM regression model: use RBF kernel with $C = 100, \gamma = 0.5$.

   ii. Random forest: use 3 features to determine the optimal split in each node of the trees. Find the optimal number of decision trees between 30 and 50.

   iii. Neural network with one hidden layer: use ReLU as activation function. Choose the best number n of hidden neurons from $n = 8, 16, 32, 64, 128$. For other parameters, feel free to pick your preferred ones.

b) Find the 5-fold cross validation mean score for each of the trained models in (a).

We would like to know how these models predict the payoff when time to expiry is zero.

c) Create a test dataset with 251 examples where the spot price ranges from 50,51,52,… 300 and all have the same strike = 180, time to expiry = 0 and option volatility = 0%. Predict the option prices for the test dataset using the 3 trained models.

d) Plot the predicted price results in (c) against the spot prices. Which trained model is the poorest in predicting the payoff at expiry? Can you give a possible explanation for that?

QUESTION 3

The duration formula for a bond that has n remaining coupon payments is given as follows:

$$D = \sum_{i=1}^{n} t_i \left[\frac{CF_i e^{-yt_i}}{B} \right]$$

Where: B is the bond price

y is the yield to maturity (continuously compounded)

CF_i is the bond cashflow for the ith remaining coupon period

t_i is the time in years from value date to the ith coupon date (assuming 365 days in a year)

Below is a class that provides bond calculations.

```
class Bond:
    def __init__(self, start, maturity, coupon, freq):
        self.start = start      # start date of bond issue
        self.end = maturity     # maturity date of bond issue
        self.coupon = coupon    # annualized coupon rate in percents (eg. 5)
        self.freq = freq        # no. of periods per year (eg. 1: annual, 2: semi-annual)
```

Two class methods are already defined and available in this class.

```
def cc_yield(self, value_date, price):
    # returns yield to maturity (continuously compounded) for a given bond price input
    # input price: in percents (eg. 101.05)
    # output yield: in decimals (eg. 0.045)
```

```
def period_dates(self, value_date):
    # returns a numpy array of remaining coupon period dates after value_date
    # example:
    #    for a bond that matures on 2022-12-10, paying coupons semi-annually
    #    if value_date is '2022-02-15', then
    #    period_dates(value_date) returns an ndarray of ['2022-06-10','2022-12-10']
```

A function to compute the number of calendar days between two dates is also defined and available.

```
def days(start, end):
    # returns the number of calendar days between start and end date inputs
```

Use the above defined class methods and function to write a class method (called **duration**) for the Bond class to compute and return the duration of a bond given an input price quoted in percents. The inputs for the method are: **value_date** and **price**.

Note: you **<u>don't need to run</u>** this piece of code as the Bond class is not explicitly provided to you.

QUESTION 4

This question is about training a SVM binary classifier **without** using scikit-learn.

In class, we discussed the mathematical formulation of the SVM binary classification model which is equivalent to solving for the $\alpha_i{}'s$ in the following optimization problem.

$$\min_{\boldsymbol{\alpha}\geq 0} J(\boldsymbol{\alpha}) \text{ subject to } \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \alpha_i y^{(i)} = 0 \text{ and } 0 \leq \alpha_i \leq C \text{ for i=1,…,m}$$

Where

$$J(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{\boldsymbol{i,j=1}}^{\boldsymbol{m}} \alpha_i\alpha_j y^{(i)}y^{(j)}K\left(\boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{x}^{(j)}\right) - \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \alpha_i$$

Suppose we have a training set of 3 examples with 2 features. The examples are: $(-1,1), (0,1), (1,1)$ and their corresponding labels are $y^{(1)} = 1, y^{(2)} = -1, y^{(3)} = 1$. (eg. we can visualize them as 3 points in a 2D plane, two in red color and one in green color)

Let's say, we use $C = 100$ in the SVM binary classifier with a polynomial kernel of degree 2 defined as

$$K(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} \cdot \boldsymbol{v})^2 \text{ for any } \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{\mathbf{2}}$$

and $\boldsymbol{u} \cdot \boldsymbol{v}$ is the dot product of $\boldsymbol{u}, \boldsymbol{v}$.

a) Write down, explicitly, the function $J(\boldsymbol{\alpha}) = J(\alpha_1, \alpha_2, \alpha_3)$ and the constraints for this training problem.

b) Use (a) to find $\alpha_i{}'s$ that minimize the value of $J(\boldsymbol{\alpha})$. (Hint: write α_2 in terms of α_1, α_3)

c) In this particular case, the decision function can be written as $f(x_1, x_2) = a{x_1}^2 + b$. What are the values of a and b?