Teaching ML in Professional Education

Phillip G. Bradford* Nadia Udler†

August 15, 2025

1 Abstract

Use of ML has become mainstream in many areas of business. There is no shortage of courses offering different paths to mastering ML skills. The most popular approaches are those of Andrew Ng and Jeremy Howard. The former assumes a decent mathematical background and is essentially deductive, while the latter approach is more inductive and requires high school level of math and limited coding skills. Obviously, both are valuable in their own ways.

In the practice of continual education, however, the unique challenge is to work with people of diverse educational backgrounds and quickly - without long term commitment - bring them to a level where they can start using ML for their applications.

Due to growing demand in fast ML education many professional education departments started to offer ML classes that do not require full time commitment from a student. However, they usually require decent Python coding skills. We are different in several aspects: we focus on concepts rather than skills and require only minimal coding background. We focus on concepts that underly ML methods rather than specific ML methods. We cater to a wider student audience.

To address this challenge, we use the ideas borrowed from teaching of Operations Research (OR). After providing the students with a brief overview of ML approaches and tools, we offer them a menu of possible projects that can be tweaked to their specific goals and requirements. Among such projects are the ones based on diet, assignment, scheduling and logistic problems. Those classical problems from OR field can be tweaked to become machine learning models.

We use educational library minpy [**?**] to demonstrate essential concepts from machine learning, such as online learning.

Our methodology may be useful for instructors of ML in professional education, as well as engineering undergraduate programs. This library

*The University of Connecticut, Stamford, CT 06901, USA; phillip.bradford@uconn.edu

†Connecticut State Colleges and Universities, Norwalk, CT, USA; nadia.udler@ncc.commnet.edu

2 Introduction

The focus of the class is on basic machine learning (ML) ideas and concepts.

Our approach is borrowed from Operations Research. Topics from Operations Research have been used to teach students with elementary math skills mathematical modeling and foster interest in STEM subjects (see [31]). This differentiates us both from classes that require a background in hard sciences and from classes that fully rely on ML software and do not require a significant math knowledge from a student.

Due to growing demand in fast ML education many professional education departments started to offer ML classes (see, for example, [27, 28, 29]) that do not require full time commitment from a student. However, they usually require decent Python coding skills. We are different in several aspects: we focus on concepts rather than skills and require only minimal coding background. We focus on concepts that underly ML methods rather than specific ML methods. We cater to a wider student audience.

This approach allows students to easily acquire the terminology and continue to study independently after finishing the class - through books, videos and software documentation. The course is project-based. Each time the selection of the projects is different. It depends on the math knowledge of the students. First several projects are demonstrated to the whole class and utilize minimal mathematical background among participants. Students do not have to learn new math but see how their existing math knowledge can be used in ML. Many students already have experience with some ML methods. Our class is helpful for understanding similarities between familiar methods. The minimal mathematical background required for this class is: solving systems of linear equations and inequalities.

Topics of machine learning are grouped by required mathematical background. Relationship with other disciplines is highlighted - statistics, probability, linear algebra, operations research, optimization, financial engineering. Students work in groups. Members of the group may have different skills, and the reason is to have a variety of skills present in the group: computer programming, mathematics, mathematical modeling, subject matter expert knowledge. This allows to complete realistic projects. Educational software minpy [8] is used for many projects.

3 Projects

3.1 Definition

Demonstration projects show how to solve real life problem using machine learning approach and using ML software libraries or other scientific software libraries. Almost half of class time is devoted to demonstration projects. Students projects are structured in the same manner as demonstration projects. After watching several demonstration projects students are able to start working on

their own projects, using guidance from the instructor.

Project consists of five steps:

1. Problem setup.

2. Creating mathematical model of the problem.

A special focus is on representing the model in a form consistent with API of a software that is used to solve the problem. We encourage the use of standard software tools to reduce the development time, increase the reliability of the solution, and require less math and programming skills. Students may still choose to implement the solution from scratch , without using standard software, as a useful exercise. We use scikit-learn and scipy for most demonstrations (see [30, 32]). Implementation of optimization methods from scratch is dewmonstrated using minpy library(see [8]).

3. Preparing input data.

4. Developing a solution, writing computer code, calling existing software libraries, and testing the code.

5. Analyzing the solution.

When working in groups students may divide the responsibilities inside the group and work on subtasks in parallel: preparing data, writing code, preparing the report, etc. Depending on math background a student may be assigned a project where higher level of math is needed.

3.2 Demo Projects and self-paced projects

Demonstration projects require minimum math background and programming skills. They are based on classical examples from different fields such as operations research, data science, statistics, engineering, and most recent examples from machine learning field. and student p One example of the demo project is considered in detail here. It is based on the Diet Problem, the classical problem from Operations Research. It uses food nutrition datasets and several toy datasets. The formulation of the problem is becoming more complex gradually, allowing students to participate in the process of building mathematical model. Each modification of the model demonstrates important ML concept.

Alternatevely first demo may be building movie recommendation system based on movie dataset. Our experience shows that this is a choice for students interested in Natural Language Processing.

In self-paced projects students may utilize their subject matter expertise, advanced math and programming skills to work on real-life problems.

Project example for a group with good programming skills and college level mathematics would be Comparing Genetic Algorithm with Covariance Matrix Adaptation Evolution Strategy, based on minpy library modules.

Project example for group with good programming skills, college-level mathematics and expertise in finance would be Portfolio selection using Black-Litterman approach [33].

Project example for a group with good programming skills, college-level mathematics and special interest in optimization may be Applying proximal

operator as a step in generalized global optimization algorithm derived from potential theory. See Appendix for the description.

3.2.1 Demo project example - The Diet Problem

Diet problem is a classical problem from the field of Operations Research. It was studied extensively in 1940s, based on the needs of WWII. Linear programs for resource allocation were formulated by Leonid Kantorovich in 1939. The Diet Problem was formulated as a linear program with the objective of minimizing the cost of a daily diet of a person while fulfilling nutritional requirements. First algorithm for solving linear programs, simplex algorithm, was developed by George Dantzig in 1947. Soon after the duality theorem for linear programming was suggested by John von Neumann, extending capabilities of linear program sensitivity analysis. Diet selection demonstration on NEOS server can be found here [26]

We explore several modifications of the original problem, including the multi objective version, interactive diet selection, taking into account dietitian's suggestions such as restrictions on certain food combinations, taking into account uncertainty in the input data, etc. We create a framework based on food/nutrition dataset and several Python solvers that allow one to add algorithms and modes easily, analyze data, perform sensitivity analysis on optimization models and store data on typical diet selection and expert data. Such a decision support system can be used for individual diet recommendations and for analysis of typical diets, such as for identifying dietary patterns in certain groups of people. This system can be used by students of AI ,ML, optimization and simulation to study modeling techniques.

As advanced demonstration project we show how to solve a task of pricing new food item using ML approach, and how it is connected to dual of the Diet Problem.

Food/nutrition data can be easily understood by beginners as well as students with advanced degrees in different disciplines.

The following subproblems are useful to specify when studying ML concepts.

1. Exploring food dataset and introducing linear classsification of food items into groups.

Food/nutrition databset consists of food items with their chemical composition (including fat, protein, carbohydrates, calcium, vitamin A, vitamin C, vitamin B, magnesium, copper, etc.). Different sources list approximated values, ranges, mean values or median values of chemical elements in each food item as it is impossible to measure the composition exactly and the composition of elements may be different in the same kind of vegetable grown in different regions, or stored in different conditions. It makes sense to accommodate this uncertainty in the model for the diet problem. One of the variants we consider is minimizing the price of daily diet under conditions that nutritional needs are satisfied with certain (large) probability. It may be useful to know that allowing 0.01 percent doubt that my breakfast provides all nutritional needs would make it 10 dollars cheaper. Sensitivity analysis of the model may reveal the fact that

we do not need to run the optimization again for creating several similar diets but need to analyze the result of single optimization run. For example, shadow prices (part of sensitivity analysis output in standard optimization tool) show the change in optimal value when right hand side is changed by one unit. This may correspond to a situation when we computed a cost for a diet with maximum of 1500 calories allowed and want to know how much the 1600 calories diet cost without running the optimization tool again.

This section introduces several ML concepts such as classification, and is helpful in explaining th difference between classification and regression by comparing mathematical models that underly the former and the latter. Perceptron algorithm introduced in 1963 in [4] may be introduced here, as an example of online classification. Simpler dataset may be utilized as sub-example, such as Airplanes dataset.

Another useful and easily implemented analysis of food dataset is visualizing food items in the space of principal components, which shows clearer separation of the food items and is used for identifying dietary patterns in different groups of people (see, for example, [10]. Principal Component Analysis (PCA) is one of widely used ML techniques, implemented in modern ML software (we use Scikit learn implementation) and all classic scinetific libraries.

scipy.optimize.linprog solver is used for solving diet problem in classic formulation.

2. Historical formulation as linear program, with objective to minimize the cost of the diet and with constraints on daily nutritional requirements.

This section introduces students to mathematical modeling. Mathematical modeling is an essentail step in solving any machine learning problem. Introduction to diet problem allows to acqure these skills easily, with only needed background being intermediate algebra.

3. Ways to solve this problem with computer and without computer.

We explore different approaches: historical approach (by exhaustive search) and the simplex algorithm of George Danzig, graphical solution in two dimentions. We discuss implementation of this algorithm in modern software, interfacing with Python libraries, Python's linprog function in SciPy optimize

4. Analysis of the solution and additional constraints.

5. Modern formulations, different objectives, applying the model in different industries.

6. Dual problem formulation, analysis of the solution.

One of the interpretations of dual for the diet problem is that certain pharmacy is producing pills such as supplements of protein, fat, calcium, etc., and trying to set the prices for such pills. The pharmacy would like to maximize the price of the pill, however, this price should be in sink with the price of food items that deliver the same chemical elements. If the pill is too expensive many people would prefer to eat real food instead of a supplement. As contemporary diets often include meal replacement shakes or bars, the dual problem can be formulated as maximizing the price of a meal replacement shake. This formulation will be considered in detail.

This sections introduces the concept of duality necessary in ML models. Historically, essential machine learning tool that used the concept of duality was support vector machines (SVM), see [11].

After this sections students should be able to read original paper by Vapnik whcich introduces SVM and software documentation in modern implementations of SVM. Methodologivcally it is sometimes advisable to step back and introduce simpler model whish demonstrates duality such as Morra Game project mentioned in the project list. The only background needed for this section is intermediate algebra. Duality theory opens other ways to analyse the solution of the primal problem intuitevely, but for the purpose of our class we stop here and refer interested students ito additional self-study materials.

7. Formulating the diet problem when nutritional facts about each food item are given as ranges rather than exact numbers.

This reformulation idemonstrates the concept of probabilistic modeling. This is important in ML. It utilizes the students basic knowlege in probabity and statistics.

The following section provides detailed mathematical formulation of all subproblem. Please see [1] for more background.

We introduce the following notations:

m – number of nutrients, n – number of foods,

Vector c of unit costs for food item, where c_i is the unit cost of food i, i=1,n

Matrix A of nutrients in each food item, where a_{ij} is the amount of nutrient j in food i, i=1,n, j=1,m

Vector b of daily nutrient requirements, where b_j is the daily requiremnt for nutrient j, j=1,m

Vector X(decision vector) of portions of each food item, where x_i -numbers of servings of each food in the daily diet,

cost of daily diet:

$$c_1x_1 + c_2x_+, .. + c_nx_n = \sum_{i=1}^{n} c_ix_i \tag{1}$$

amount of nutrient j in daily diet:

$$a_{1j}x_1 + a_{2j}x_+, .. + a_{nj}x_n = \sum_{i=1}^{n} a_{ij}x_i \tag{2}$$

The objective of the problem is to minimize the cost of daily diet:

$$\min_x \sum_{i=1}^{n} c_ix_i \tag{3}$$

The constraints that amounts of nutrients j should not be less than the recommended amounts b_j are expressed as follows

$$\sum_{i=1}^{n} a_{ij}x_i \geq b_j, j = 1, m \tag{4}$$

The bounds on decision variables (constraining number of portions for each food item) are expressed as follows:

$$x_i \geq 0, i = 1, n \tag{5}$$

It is a good idea to introduc matrix form of the problem as well, for better understanding of how Python tools for solving such problems (the linprog function from scipy.optimize) are used.

The dual of the diet problem may be interpreted as problem of revenue maximization so that artificial foods are competitive with the real foods in price, where the variable $U = (u_1, ..u_n)$ is interpreted as the unit prices of nutrient pills. Or, think of A pharmaceutical company which is developing the pills for meal replacement and trying to maximize its profit under the requirement that the price paid for the nutritional value found in the food item (but replaced by a pill) is not exceeding the cost of the food item.

Revenue maximization objective:

$$\max_u \sum_{j=1}^{n} b_j u_j \tag{6}$$

Constraints :

$$\sum_{j=1}^{m} a_{ij} u_j \leq c_i, i = 1, n \tag{7}$$

The bounds on decision variables:

$$u_j \geq 0, j = 1, m \tag{8}$$

Now consider some modifications of the original diet problem.

Many diets do not allow to consume certain food combinations in one meal: for example, this is how to model a constraint on two types of animal protein consumption in the same meal: for each food item i a binary variable z_i is introduced. $z_i = 0$ if food item i is not an animal protein, and $z_i = 1$ if item i is an animal protein.

The new constraint is

$$\sum_{i=1}^{n} z_i = 1 \tag{9}$$

This constraint makes the problem integer linear, which may require different solver.

Next we consoder probabilistic formulation.

Because chemical composition of food items are not known exactly (represented by matrix A), we assume normal distribution of values for each nutrient in each food item, with mean and standard deviation found in food dataset. We solve the following deterministic equivalent of stochastic optimization problem:

Now A with elements a_{ij} is a matrix of random variables where elements represent amounts of nutrient j in food i, and a_{ij} is normally distributed with mean $\mu_i j$ and standard deviation $\sigma_i j$.

Constraints in the original problem are modified to accomodate uncertainty: probability that requirement for nutrient j is satisfied is greater than certain probability p

$$P(-\sum_{i=1}^{n} a_{ij}x_i \leq -b_j) > p, j = 1, m \tag{10}$$

This is a chance-constrained problem but can be transformed to linear program as follows:

Note that linear combination of independent (assumed) normally distributed random variables $\sum_{i=1}^{n} a_{ij}x_i$ is normally distributed with mean $\sum_{i=1}^{n} \mu_{ij}x_i$ and standard deviation $\sqrt{(\sum_{i=1}^{n} \sigma_{ij}^2 x_i^2)}$

Denote the quantile of standard normal distribution by q we get:
$-\sum_{i=1}^{n} \mu_{ij}x_i - q\sqrt{(\sum_{i=1}^{n} \sigma_{ij}^2 x_i^2)} \leq -b_j$
$\sum_{i=1}^{n} \mu_{ij}x_i + q\sqrt{(\sum_{i=1}^{n} \sigma_{ij}^2 x_i^2)} \geq b_j$

We used Python linear optimization tool from ScipPy library (scipy.optimize.linprog) for solving classical formulationof the diet problem.

```
res=linprog(c,A_ub=A_ub,b_ub=b_ub,bounds=bounds)
```

The complete example of solving diet problem can be found here [25]

If nonlinear constraints are added the nonlinear solver from scipy library can be used. The new interface for optimization (function minimize) allows to specify constraints, objective function, and solution method)

3.2.2 Advanced project examples

3.2.3 Applying proximal operator as a step in generalized global optimization algorithm derived from potential theory

Required background: Python programming, college-level math

Machine learning models are often provided as functions in software packages. Optimization of such functions is a challenging task because traditional derivative based optimization methods with guaranteed convergence properties cannot be used. In [8] we introduced educational software library minpy that allows to create new optimization methods with desired properties. Those methods include elementary modules designed using the approach for constructing global optimization algorithms based on potential theory [3, 2]. They do not use derivatives of objective function and as a result work with nondifferentiable functions (or functions given by computer programs, or black box functions), but have guaranteed convergence. The focus of the library is on the transparency of the generalized optimization method rather than on efficiency of

implementation. This is important since this library is intended to be used for educational purposes. To further generalize the optimization algorithms, we suggest replacing certain steps of the algorithms with proximal operator applied to the current point of the search. This does not compromise the convergence of the algorithm [3] but allows to write the algorithm in more concise manner, making it more transparent. Suggested reading to learn more about proximal operators and proximal algorithms: [7, 5, 6]

3.2.4 Fine Tuning LLM for Specific Knowledge Area

Required background: Python programming, college-level math

Large Language Models (LLM) are used in many applications in social and natural sciences, law, education, etc. LLMs require large amount of data for initial training, and then they can be fine-tuned on recent data, Such fine-tuning is perfomred using online or continual optimization algorithms. Such algorithms have a common problem - they are prone to catastrophic forgetting. Optimization algorithm from our module minpy is designed based on Kaplinskiy's potential functions approach.

These algorithms were inherently designed to operate in an online mode, where data arrives sequentially and parameter updates occur continuously. One significant advantage of this approach is its ability to mitigate the effects of catastrophic forgetting—a common issue in neural networks where previously learned information is lost when new data is introduced.

Read more on this approach in a separate document.

In this project students explore the behaviour of optimization algorithm from minpy library and its ability to overcome catastrophoc forgetting.

3.2.5 Hallusinations in LLM

Required background: Python programming, college-level math

Language Modeling has become a neccesary part of ML education. This project explores NLP situations modeled as optimization problems. As LLMs are used for many applications, their shortcomings are explored and addressed. Optimization plays important role in such tasks. For example, see [9] for dealing with LLMs hallucinations using discrete optimization algorithm.

In this project students are suggested to start with simple discrete optimization algorithm - branch-and-bound method, and then introduce additional parametrs, randomization, etc, to make it more adaptable. bb method from minpy library may be used.

3.2.6 Introduction to neural network optimization

Required background: Python programming, college-level math

This project introduces most important concepts in artificial neural networks. Students implement the perceptron algorithm with linear kernel - famous online classification method introduced in [4], and explore differences and

similarities of this method to SVM. Educational library minpy [8] is used to implement neural network optimization approaches described in [13, 14, 15]. As a real life application of ANN the problem of pricing new product in food industry is considered. Many factors affect food prices directly or indirectly, such as agricultural productivity, government policies, overall market state, etc.We do not have exact formula to express dependency for computing prices knowing values of factors, besides, there are many unknown or hidden factors.. Such problem is a good candidate for solution by ML methods, such as ANN.We use data (Nutritional content and prices of different foods) to train the model and then use it to assign a price to a new product, such as energy drink, given its nutritional content. Full tutorial on this task is given in supporting Jupyter notebook [34].

Prerequisite for this project: familiarity with diet optimization problem

3.2.7 Restricted Boltzmann Machine (RBM)

RBM can be thought of as maximum likelihood estimation of parameters of certain probability function.. This probability function is represented by feed forward neural network with one hidden layer and one input (visible) layer, and neurons from each layer are connected to all neurons in other layer but not connected to each other. Learning is performed using special fast algorithm - Contrastive Divergence [17], [18]. Furthre improvent in learning efficiency is suggested in this paper [16] and in this dissertation [23]. RBMs are often used in methods for discovering patterns in data, such feature extraction, and nonlinear dimentionality reduction [19] . For application in finance see this paper [20] and dissertations [21, 22].

Prerequisite for this project: familiarity with Maximum Likelihood Estimation and Expectation Maximization Algorithm,Feed-Forward Neural Networks.

3.2.8 Black Literman Model for portfolio selection and online learning

RBM can be thought of as maximum likelihood estimation of parameters of certain probability function.. This probability function is represented by feed forward neural network with one hidden layer and one input (visible) layer, and neurons from each layer are connected to all neurons in other layer but not connected to each other. Learning is performed using special fast algorithm - Contrastive Divergence [17], [18]. Furthre improvent in learning efficiency is suggested in this paper [16] and in this dissertation [23]. RBMs are often used in methods for discovering patterns in data, such feature extraction, and nonlinear dimentionality reduction [19] . For application in finance see this paper [20] and dissertations [21, 22].

Prerequisite for this project: familiarity with Maximum Likelihood Estimation and Expectation Maximization Algorithm,Feed-Forward Neural Networks.

References

[1] Robert J. Vanderbei (2014) *Linear Programming: Foundations and Extentions*,Springer

[2] Kaplinskii A.I., Propoi A.I. (1994) *First-order nonlocal optimization methods that use potential theory.*, Automation and Remote Control.

[3] Kaplinskij, A.I., Pesin, A.M., Propoj, A.I.. (1994) *Analysis of search methods of optimization based on potential theory. III: Convergence of methods*, Automation and Remote Control

[4] Aizerman, M. A., Braverman, Emmanuel M., Rozonoer, L. I. (1964) *Theoretical foundations of the potential function method in pattern recognition learning*, Automation and Remote Control

[5] Rémi Gribonval, Mila Nikolova (2020) *A characterization of proximity operators*, Classical Analysis and ODEs

[6] Zan Chen, Wenlong Guo, Yuanjing Feng, Yongqiang Li, Changchen Zhao, Yi Ren, Ling Shao (2021) *Deep-learned regularization and proximal operator for image compressive sensing*, IEEE Transactions on Image Processing

[7] Lingxiao Li, Noam Aigerman, Vladimir G. Kim, Jiajin Li, Kristjan Greenewald, Mikhail Yurochkin, Justin Solomon (2023) *Learning Proximal Operators to Discover Multiple Optima*

[8] Nadia Udler (2014) *Global optimization software library for research and education*,Scipy Conference Proceedings

[9] Erik Jones, Anca Dragan, Aditi Raghunathan, Jacob Steinhardt (2023) *Automatically Auditing Large Language Models via Discrete Optimizationa*

[10] Carolina Schwedhelm, Khalid Iqba, Sven Knüppel, Lukas Schwingshackl, Heiner Boeing (2018) *Contribution to the understanding of how PCA–derived dietary patterns emerge from habitual data on food consumption*,Clinical Nutrition

[11] Cortes Corinna, Vapnik Vladimir (1992) *Support Vector Networks*,Machine Learning

[12] Alexander I. Galushkin (2007) *Neural Networks Theory*,Springer

[13] Yufei Cui, Ziquan Liu, Wuguannan Yao, Qiao Li, Antoni B. Chan, Teiwei Kuo, Chun Jason Xue (2020) *Fully nested neural network for adaptive compression and quantization*

[14] Yufei Cui, Ziquan Liu, Wuguannan Yao, Qiao Li, Antoni B. Chan, Tei-wei Kuo, Chun Jason Xue (2021) *Bayesian Nested Neural Networks for Uncertainty Calibration and Adaptive Compression*

[15] Kaplinskij, A.I., Pesin, A.M (2018) *Fast converging optimization methods for multi layer perceptrons*,Vestnik, the Journal of Voronezh University

[16] Ferran Mazzanti and Enrique Romero (2020) *Efficient Evaluation of the Partition Function of RBMs with Annealed Importance Sampling*

[17] Geoffrey Hinton (2010) *A Practical Guide to Training Restricted Boltzmann Machines*

[18] Ilya Sutskever, Tijmen Tieleman (2010) *On the Convergence Properties of Contrastive Divergence*

[19] Guy Bresler, Frederic Koehler, Ankur Moitra, Elchanan Mossel (2018) *Learning Restricted Boltzmann Machines via Influence Maximization*

[20] Carlos Assis, Adriano C. M. Pereira, Eduardo G. Carrano (2018) *Restricted Boltzmann Machines for the Prediction of Trends in Financial Time Series*

[21] Gustav Fredriksson, Anton Hellstrom (2019) *Restricted Boltzmann Machine as Recommendation Model for Venture Capital*

[22] João Henrique, Fialho Rodrigues (2019) *Time series analysis using restricted Boltzmann machines and dynamic Bayesian networks*

[23] Rares-Darius Buhai (2019) *Learning Restricted Boltzmann Machines with Few Latent Variables*

[24] Food Dataset, *'https://catalog.data.gov/dataset/supertracker-source-code-and-foods-database/resource/bb32cdd7-7c95-4a7a-99ff-4c5c578ad33c'*

[25] Diet Problem Tutorial, *'https://www.github.com/nadiakap/PythonClass/DietProblem/'*

[26] NEOS Server, *'https://neos-guide.org/case-studies'*

[27] MIT, (2023) *https://professionalonline2.mit.edu/no-code-artificial-intelligence-machine-learning-program*

[28] GMU, (2023) *https://cpe.gmu.edu/courses-and-programs/courses/gsa-cptt/gsa-cptt-lhl-0230-machine-learning.php*

[29] University of Utah, (2023) *https://bootcamps.continue.utah.edu/micro/artificial-intelligence/*

[30] Virtanen, Pauli et al (2020) *Scipy:Fundamental Algorithms for Scientific Computing in Python*,Nature Methods

[31] Raffaele, A., Gobbi, A.(2021) *Teaching Operations Research Before University: A Focus on Grades 9–12*,Operations Research Forum

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.(2011) *Scikit-learn: Machine Learning in Python*,Journal of Machine Learning Research

[33] Ziye Yang, Ke Lu (2025) *Enhancing Black-Litterman Portfolio via Hybrid Forecasting Model Combining Multivariate Decomposition and Noise Reduction*

[34] Pricing new foods (2023)*https://www.github.com/nadiakap/PythonForML*