# Mini Project

## Due: Fri, December 12, 2025 at 11:59pm ET

## Note: Late Submissions will NOT be accepted

Up until now, we have given you fairly detailed instructions for how to design data analyses to answer specific questions about data – in particular, how to set up a particular analysis and what steps to take to run it. In this project, you will put that knowledge to use!

Put yourself in the shoes of a data scientist being given a data set and asked to draw conclusions from it. Your job will be to understand what the data is showing you, design the analyses you need, justify those choices, draw conclusions from running the analyses, and explain why they do (or do not) make sense.

We are deliberately not giving you detailed directions on how to solve these problems, but feel free to attend office hours to brainstorm.

## Partners

On this project **you are allowed to work with one partner from the same section**. Working with a partner is **optional**, and working with a partner will not impact how the project is graded. If you want to work with a partner, it is your responsibility to pair up; feel free to use Piazza's "Search for Teammates" feature to facilitate this. **Note again that you can only team-up with a person from the same section.**

## Submission

You will submit any code that you write and your final PDF report to Gradescope. There are no direct guidelines on the code, except we expect your code in the submission to run if the library dependencies are satisfied.

It is highly essential to note that, if you are planning to work with a member, then **only the team lead, i.e., one of the two members (decided among yourselves), needs to submit the completed code and the PDF report on Gradescope. Do not submit a zip/archive and make sure you submit a PDF report (NOT a docx or other format).** The points on this project obtained by the team leader will be reflected on their teammate as well. **To make sure the scores match among the teammates we request you to utilize Gradescope's group submission feature**. If you decide to work solo, then you are your group's leader.

## Objectives

There are three possible paths through this project:

1. You may use dataset (path) #1, which captures information about student behavior and performance in an online course. See below for the analysis questions we want you to answer.
2. You may use dataset (path) #2, which captures information about bike usage in New York City. See below for the analysis questions we want you to answer.
3. You may use dataset (path) #3, which are images of digits. See below for the analysis questions we want you to answer.

## Path 1: Student performance related to video-watching behavior

`behavior-performance.txt` contains data for an online course on how students watched videos (e.g., how much time they spent watching, how often they paused the video, etc.) and how they performed on in-video quizzes. `README-behavior-performance.pdf` details the information contained in the data fields. There might be some extra data fields present than the ones mentioned here. Feel free to ignore/include them in your analysis. In this path, the analysis questions we would like you to answer are as follows:

*You will run prediction algorithm(s) for **ALL** students for **ONE** video, and repeat this process for all videos.*

1. How well can the students be naturally grouped or clustered by their video-watching behavior (`fracSpent`, `fracComp`, `fracPaused`, `numPauses`, `avgPBR`, `numRWs`, and `numFFs`)? You should use all students that complete at least five of the videos in your analysis. `Hints: KMeans or distribution parameters(mean and standard deviation) of Gaussians`
2. Can student's video-watching behavior be used to predict a student's performance (i.e., average score `s` across all quizzes)? Will adding the cluster information (e.g. which group they belong) help the model improve the performance and makes the model become more/less under/overfit? Explain your conclusion and discuss the reasons why such results could happen.
3. Taking this a step further, how well can you predict a student's performance on a *particular* in-video quiz question (i.e., whether they will be correct or incorrect) based on their video-watching behaviors while watching the corresponding video? You should use all student-video pairs in your analysis.

## Path 2: Bike traffic

The `nyc_bicycle_counts_2016.csv` gives information on bike traffic across a number of bridges in New York City. In this path, the analysis questions we would like you to answer are as follows:

1. You want to install sensors on the bridges to estimate overall traffic across

all the bridges. But you only have enough budget to install sensors on three of the four bridges. Which bridges should you install the sensors on to get the best prediction of overall traffic?

2. The city administration is cracking down on helmet laws, and wants to deploy police officers on days with high traffic to hand out citations. Can they use the next day's weather forecast(low/high temperature and precipitation) to predict the total number of bicyclists that day?
3. Can you analyze and visualize the data to identify any patterns or trends associated with specific days of the week? (Hint: One way is that you can average the values over all weekdays and then see if there are some weekly patterns.) Can you use this data to predict what *day* (Monday to Sunday) is today based on the number of bicyclists on the bridges?

## Path 3: Data Security in Model Training

The first task is to understand the digits dataset from sklearn. This dataset information is found at sklearn.datasets.load_digits — scikit-learn 1.1.3 documentation. You may find this link helpful: load_digits. There are 10 classes (0-9) and each datapoint is a 8 x 8 image of a digit. The first task is to understand this data by importing the dataset and printing out some of the samples. You will need to do the following in Path 3:

1. Complete the code for `dataset_searcher` and `print_number` in `MiniProjectPath3.py`
2. print out and plot the numbers of the class `[2, 0, 8, 7, 5]`

Incomplete sample code is given as a guideline to create and fit different models to the data. Please refer to the sklearn documentation of (1) GaussianNB, (2) KNeighborsClassifier, and (3) MLPClassifier, to properly do the following:

3. Get the predicted values of the model with the **Test data** with the `GaussianNB` model
4. Calculate the overall accuracy of the model (out of the predicted labels, how many were correct?) by finishing the definintion of `OverallAccuracy` and find the value for the overall accuracy of the `GaussianNB` model
5. Get the predicted values and show the results of the model with the numbers `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]` (use instead of `X_test`) with the `print_number` function
6. Repeat steps 3 and 5 with the `KNeighborsClassifier` and the `MLPClassifier`
7. Discuss your results. Is there a difference between the performance of the three models?

Now some of the training data is "poisoned." This is shown in the later part of the `MiniProjectPath3.py` code.

8. Describe what is happening to the training data.

9. Repeat steps 3-6, but this time, use the poisoned training data to fit the model. Note that the evaluation still should be done for the **test data**.
10. Discuss how the three model performances have changed after poisoning the training data.
11. Discuss what model showed strongest robustness against the poisoning.

Now try to "denoise" the "poisoned" training data using denoising functions provided from the sklearn library. We suggest you to use the `KernelPCA` method for denoising the corrupted training data. The following link can be helpful: KernelPCA

12. Describe what is happening to remove the noise.
13. Discuss how Poison Data 1 differs from the denoised data.
14. After denoising the training dataset, repeat steps 3-6 with the denoised training data. Note that the denoised training data is used only for fitting the model, and the evaluation should be done for the **test data**.
15. Discuss how the three model performances have changed after applying the denoising steps. Have the performances improved? Or was there no significant difference?

## What to turn in

You must turn in two sets of files, code and a report PDF, both to be submitted to **Gradescope**:

- `report.pdf`: A project report, which should consist of the following sections. Take a look at `MiniProjectReportTemplate.docx` on a rough template for the report. **Submit your report as a PDF file.**
  - A section with the names of the team members (maximum of two), your Purdue username(s), and the path (1 or 2 or 3) you have taken.
  - A section describing the dataset you are working with.
  - A section describing the analyses and methods you chose to use for each analysis question (with a paragraph or two justifying why you chose that analysis and what you expect the analysis to tell you).
  - A section (or more) describing the results of each analysis, and what your answers to the questions are based on your results. Visual aids are helpful here to back up your conclusions. **Note that, it is OK if you do not get "positive" answers from your analysis, but you must explain why that might be the case**.
- All Python `.py` code files you wrote to complete the analysis steps.