Project 4: Race with Bayes

MTH448

Due: 4/8/2025

1 Introduction

In this project you will perform analysis on a dataset of the marathon results of 26,000 runners. We will attempt to predict the gender of runners from their finishing time by using probability, specifically Bayes' Theorem.

1.1 Bayes' Theorem

Bayes' theorem states that for events A and B the probability of event A occurring given event B occurred ($P(A|B)$) can be computed using the equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

$P(A\|B)$	Probability of A occurring given B occurred.
$P(B\|A)$	Probability of B occurring given A occurred.
$P(A)$	Probability of A occurring whether or not B occurred.
$P(B)$	Probability of B occurring whether or not A occurred.

2 Project Components

1. Split the data, randomly, into testing and training data.

2. Compute 1-dimensional kernel density estimate (KDE) of male and female runners in the training data using finish times, and use it together with the Bayes theorem to compute the probability that a runner with a given time was a female. Use this to make predictions if a runner was a male/female based on their finish times and check accuracy of these predictions for the testing data.

    - Consider different kernel bandwidths and find which bandwidth value produces the best prediction accuracy. How does the dafualt bandwidth value do in comparison?

3. Repeat part 2, but using 2-dimensional KDEs computed using finish times and ages of runners (you can just use the default kernel size for this part). How was your prediction accuracy affected?

4. Compare accuracy of predictions obtained in parts 2 and 3 to the predictions made using k-NN with the same input data.

5. Add anything else that you find relevant and interesting.

Note: Tools for computing KDE are implemented by several Python libraries. You can use, for example, *scipy.stats.gaussian_kde* which is a part of the *scipy* library. On the other hand, you must not use ready-made Bayes classification tools implemented in *sklearn* and other machine learning packages. You can use *sklearn* to compute k-NN classification.

3 Bonus (up to 5 pts)

The results for the most rescent NYC marathon can be found at `https://www.nyrr.org/tcsnycmarathon/results/race-results`. Use your knowledge of requests and Beautiful soup to scrape the top 10 finishers and create a dataframe with the runners'

- Name
- Gender
- Age
- Place
- Finishing Time

4 Grading

The report grade will have the following distribution:

Element	**Weight**	**What will be graded**
Introduction Section	10%	Quality of narrative.
Conclusion Section	10%	Quality of narrative.
Report Content	30%	Work done on developing the project. Your analysis, insights, observations, and interpretations. Quality of the narrative of the report.
Python Code	30%	Quality of the Python code included in the report. Relevance of the code to the project. Code organization and readability. Documentation of code by code comments.
Presentation	20%	Organization of the report. Text formatting. Use of LaTeX to typeset mathematics. Formatting of code output. Quality of graphs and plots.

4.1 Introduction

The introduction is the first section of your report. It describes the project (the underlying math) and the goals of the report. It should be written in a way that is engaging and understandable to a student who has some background in math and coding but does not take this course. The introduction SHOULD NOT be in list form. While some of the concepts in the introduction will be repeated from the project description, you should state the concepts in your own words and not copy from the project description.

4.2 Conclusion

This section should summarize your results and major findings. It can also include potential future extensions of the project. This should be the last section of the report.

4.3 Content and presentation

Outside of the two section mentioned your report should be broken up into sections and subsections in such a way as to maximize readability and comprehensibility. Where appropriate use Latex to format math.

Please note, projects ARE NOT just sets of coding exercises. They are reports in which you explore a particular math problem or phenomenon. Projects should be readable and interesting for a person not taking this class (and thus not having access to the project description) but familiar with math and python.

4.4 Code

Your code should be readable and understandable. Code should be broken up into small snippets each of which gets its own cell. Words should be used to describe what the code is doing and what the logic of your approach is. DO NOT put all of your code into a single section with no words or discussion.

Code should be readable with understandable variable names and comments included to explain what is not clear.

All python code included in the report should be written in such a way that it can be executed sequentially. For example, a function should never be used prior to its definition.

All code included must work. Do not submit code with errors.

Code output must serve the narrative of your report and should be formatted for easy reading and understanding.

4.5 Use of external resources

You are allowed but not at all required to consult resources outside of what is presented in the course. If you use an external resource in a significant way (not just googling an error message or a command that you forgot) please include a citation in your report. You are welcome to use features of python that were not shown in this class but you must understand fully what the feature does. The instructor reserves the right to ask you to explain any fragment of your code. An inability to do so may result in significant grade reduction or even more extreme consequences.

4.6 Collaboration

While students are allowed (and even encouraged) to collaborate and discuss projects the final submission must be the work of solely the submitting individual. Any assignment that is suspected of not being the student's own work will receive a zero and further disciplinary actions may also follow if they are deemed necessary. Any use of generative AI (e.g., ChatGPT) is prohibited in this class and will be considered a violation of UB's academic integrity policy.