Drew Lickman

CSCI 4820-001

Project #3

Due: 10/9/24

A.I. Disclaimer: Work for this assignment was completed with the aid of artificial intelligence tools and comprehensive documentation of the names of, input provided to, and output obtained from, these tools is included as part of my assignment submission.

Overall, I was confused with this project due to vagueness of instructions and having to figure out how to use the sklearn library. Also, I didn’t even realize the imdb\_reviews document included the positive/negative sentiment included at the end of each line until I was almost done with the project and had to refactor a decent amount of code to fix it.

I started by using Claude 3.5 Sonnet to help explain the project to me, which helped a little. A screenshot of a computer

Description automatically generatedA black and white list of a movie review

Description automatically generated

I started by writing my custom Logistic Regression class, but got stuck with the fit() and predict() functions, as I was just really confused about what they should do and how. It confirmed that my predict function was correct, and helped significantly with the fit() function.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

I was also confused with the X features table and Y labels list. After having it explained step-by-step I started feeling a bit better about understanding the project.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

I came back to the gradient descent part of the code that Claude 3.5 Sonnet provided so that it could explain it so I could understand it better, which allowed me to adjust it to work a bit better.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

As I was testing my code and using debug tools, I discovered that my predictions were using 1d arrays, rather than 2d arrays after the first iteration. I did some bug tracking and learned that math.pow(math.e), -z)) was only working on single elements, but as suggested by Claude and using np.exp(-z)) it allowed the code to work on entire arrays at once.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedSince we used the sklearn library for the logistic regression classifier, I figured it would have a built-in function for randomizing training / test data. I asked Claude about the sklearn functions and discovered model\_selection.train\_test\_split(). I already had most of the trial loop code as skeleton code, I wasn’t sure how to use the library functions to run the sklearn model.

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated