Report for final

Link to the data set:
https://www.kaggle.com/datasets/rakkesharv/spotify-top-10000-streamed-songs

modules.rs

```
pub fn read_csv(file_path: &str) -> Result<(Vec<f64>, Vec<f64>), Box<dyn Error>> {
```

This function reads a CSV file, extracting the number of days since release (x) and total streams (y). It handles missing or invalid data by defaulting values to 0, ensuring the program runs smoothly and prepares the data for analysis.

```
pub fn normalize(data: &Vec<f64>) -> (Vec<f64>, f64, f64) {
```

This function normalizes a vector to have a mean of 0 and a standard deviation of 1. It ensures the data is on a consistent scale, which is essential for accurate gradient descent calculations in linear regression.

```
pub fn linear_regression(
```

This function calculates the best-fit line using gradient descent. It iteratively adjusts the slope (m) and intercept (b) to minimize the error between predicted and actual values, modeling the relationship between days and total streams.

```
pub fn coefficient_of_determination(
```

This function computes the R2R^2R2 value, which measures how well the regression line fits the data. A higher R2R^2R2 indicates a better fit and helps evaluate the accuracy of the linear regression model.

main.rs

The main function uses read_csv to load the data and extract the number of days and total streams. Then it normalizes the data using normalize to make the values easier to work with. After that, linear regression is run to calculate the slope and intercept of the best-fit line. Finally, it uses coefficient_of_determination to check how well the line fits the data and prints all the results, including the equation and R2R^2R2 value.

Tests Explanation

Test for normalize: This test checks if normalize works correctly by calculating the mean and standard deviation of a small dataset. It makes sure the normalized data has the right properties, like having a mean close to 0.
Test for linear_regression: This test makes sure linear_regression finds the correct slope and intercept for a dataset with a clear linear pattern. It’s a way to confirm that the gradient descent logic is working

Test for coefficient_of_determination: This test checks if R2R^2R2 is calculated correctly. For a perfect linear dataset, the test ensures R2R^2R2 is very close to 1 meaning the regression line explains the data well

Explanation of the Output

Slope (m): 0.9283

This value represents the rate at which total streams increase for every additional day since the release of a song. A slope of 0.9283 means that on average the total streams increase by approximately 0.93 units a day since the song has been out.

Intercept (b): 0.0000

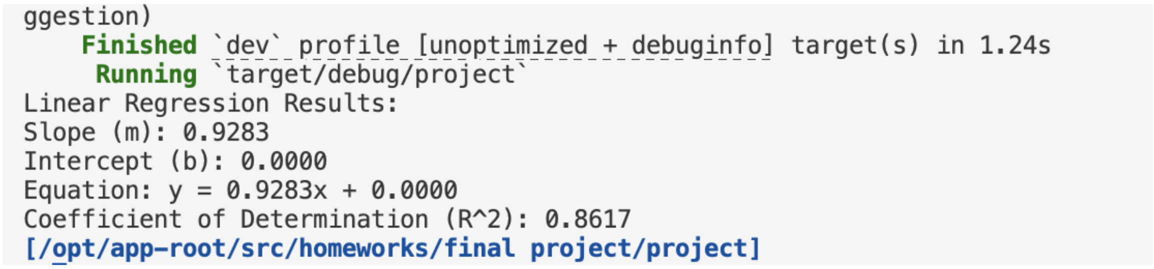
The intercept is the predicted number of total streams on the day the song was released (Day 0). The value is 0.0000 becasue the model predicts no streams on the release day.

Equation: y = 0.9283x + 0.0000

This equation represents the linear relationship between the days since **release x** and the total **streams y**. It shows how the model predicts streams based on the days since the release.

Coefficient of Determination (R^2): 0.8617

The R2R^2R2 value of 0.8617 indicates that approximately 86.17% of the variance in total streams is explained by the number of days since release. This is a strong fit meaning the number of days since release is an important factor in predicting total streams.



```
ggestion)
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 1.24s
     Running `target/debug/project`
Linear Regression Results:
Slope (m): 0.9283
Intercept (b): 0.0000
Equation: y = 0.9283x + 0.0000
Coefficient of Determination (R^2): 0.8617
[/opt/app-root/src/homeworks/final project/project]
```