DS210 Project Report

Name: Khoa Nguyen

DS210 Project

Dataset: hanoiweather.csv

Method to Analyze: Logistic Regression, built from scratch.

A. Project Overview

Goal: Predict daily rainfall (binary: > 0mm or not) in Hanoi using same-day weather features.

Dataset: *hanoiweather.csv*. ~12500 observations, 32 variables

Features Used: Temperature (*temp*), Humidity (*humidity*), Wind Speed (*windspeed* mapped to *wind*), Solar Radiation (*solarradiation* mapped to *sunshine*).

Target Variable: *rain* (*u8*, 0 or 1), derived from *precip* (mm).

B. Data Processing

Loading: *csv* and *serde* crates read *hanoiweather.csv*.

Cleaning/Transformations:

- Parsing errors skip rows.
- Empty numeric fields treated as *0.0*.
- Binary target *rain* derived from *precip*.
- Selected features extracted.
- Data randomly shuffled (*rand* crate).
- Features scaled to [0, 1] using custom *MinMaxScaler*.

C. Code Structure

Modules:

- *main.rs*: Orchestrates workflow.
- *data.rs*: Handles loading and parsing.
- *model.rs*: Implements Logistic Regression and scaling.
- *tests.rs*: Unit tests.

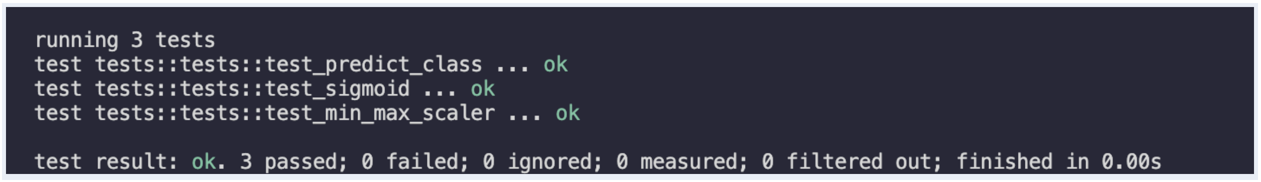
<u>Key Functions & Types:</u>

- *data::WeatherRow* (struct): Represents a CSV row.
- *data::load_weather_data*: Reads/parses CSV.
- *model::MinMaxScaler* (struct): Scales features.
- *model::sigmoid*: Logistic function.
- *model::predict_probability*: Calculates probability.
- *model::predict_class*: Predicts binary class.
- *model::train_logistic_regression*: Trains model via Batch Gradient Descent.

Main Workflow (*main.rs*): Load → Shuffle → Split → Scale → Train → Predict → Evaluate → Print Accuracy.

D. Tests

cargo test Output:



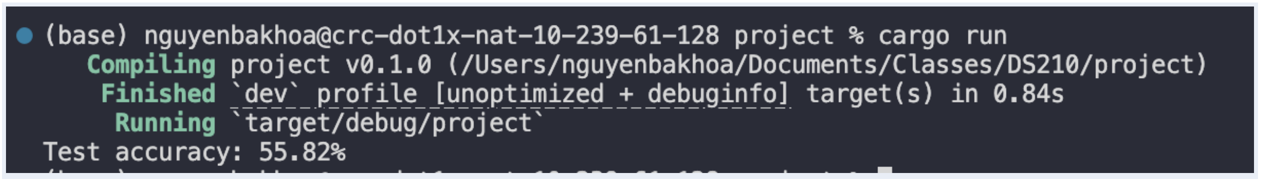
```
running 3 tests
test tests::tests::test_predict_class ... ok
test tests::tests::test_sigmoid ... ok
test tests::tests::test_min_max_scaler ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Test Descriptions:

- *test_sigmoid*: Verifies sigmoid function behavior.
- *test_predict_class*: Checks binary prediction thresholding.
- *test_min_max_scaler*: Confirms scaler functionality.
- Note: Test for *train_logistic_regression* is missing.

E. Results

cargo run Output:



```
● (base) nguyenbakhoa@crc-dot1x-nat-10-239-61-128 project % cargo run
     Compiling project v0.1.0 (/Users/nguyenbakhoa/Documents/Classes/DS210/project)
      Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.84s
       Running `target/debug/project`
  Test accuracy: 55.82%
```

Interpretation: Model achieved 55.82% accuracy on test data, performing slightly better than random chance (50%).

F. Usage Instructions

- Build: *cargo build* (or --*release*).
- Run: *cargo run* (or --*release*).
- Test: *cargo test*.
- Arguments: None.
- Expected Runtime: Few seconds.

Jupyter notebook *data_facts.ipynb* included for basic data analysis/visualization (histograms). Run cells with Ctrl + Enter/Shift + Enter.

G. AI-Assistance Disclosure and Other Citations

Used Google Gemini 2.5 Pro for:

- Understanding Logistic Regression logic/components.
- Code structure/logic feedback and optimization.
- Learning Rust crate usage (*rand*, *csv*, *serde*).
- Debugging.
- Write and structure this report better