Freelancer Data Analysis and Prediction Project

A. Project Overview

Goal:
This project aims to extract actionable insights from a freelancer dataset, focusing on uncovering patterns between freelancers' skillsets, experience, and their payoffs (earnings and hourly rates). The analysis includes clustering, statistical summaries, and regression-based prediction of hourly rates based on various features.

Dataset:
Source:
https://www.kaggle.com/datasets/shohinurpervezshohan/freelancer-earnings-and-job-trends
Size: 193.65 kB, 15 columns, and 1950 datapoints.
Fields Used:
id, job_category, platform, experience_level, client_region, earnings_usd, hourly_rate, job_success_rate

B. Data Processing

Loading:
The dataset is loaded from a CSV file using a custom Rust module (data_loader.rs).
Each row is parsed into a Freelancer struct, with error handling for missing or malformed data.

Cleaning/Transformations:
Categorical variables (e.g., job category, platform, experience level, client region) are encoded for analysis and regression.
Numerical fields (e.g., job success rate) are normalized for regression.

C. Code Structure

Modules:

data_loader.rs
Purpose: Load and parse the freelancer CSV data into Rust structs.
Key Types:
Freelancer struct: Represents a single freelancer's data.
Key Functions:
load_freelancers_from_csv(path: &str) -> Result<Vec<Freelancer>, Box<dyn Error>>
Inputs: File path
Outputs: Vector of Freelancer structs

algorithms.rs
Purpose: Implements algorithms for clustering and relationship analysis.
Key Functions:
find_connected_components(): Finds clusters of freelancers based on shared attributes. This algorithm looks for connected components in a graph by implementing the Breadth-First Search (BFS) method.
build_collaboration_graph(): Builds a graph in adjacency list where nodes are freelancers and edges represent shared jobs or attributes.
find_shared_attributes(): Identifies commonalities between freelancers.

analysis.rs
Purpose: Statistical analysis and visualization of clusters.
Key Functions:
analyze_cluster_performance(): Prints average earnings and hourly rates per cluster.
analyze_cluster_profiles(): Prints dominant job categories, platforms, regions, and experience levels per cluster.
plot_cluster_experience_rates(): Generates a bar chart of hourly rates by experience level for each cluster using Plotters.

regression.rs (Part 2)
Purpose: Multivariate regression for predicting hourly rates.
Key Functions:
run_regression(): Fits a linear regression model using Linfa, returns coefficients and intercept.

main.rs
Purpose: Demonstrates the workflow: loads data, runs clustering, does analysis, performs regression, and plots graph.

Main Workflow:

Part 1

- Data Loading: main.rs calls data_loader::load_freelancers to load the dataset.
- Clustering: algorithms module builds a graph and finds clusters of similar freelancers.
- Analysis: analysis module computes and prints statistics for each cluster.
- Visualization: analysis::plot_cluster_experience_rates creates a bar chart of hourly rates by experience level and cluster.

Part 2

- Data Loading: main.rs calls data_loader::load_freelancers to load the dataset.
- Regression: Use the regression module to fit a model to predict hourly rates from features.

D. Tests

Part 1:

Cargo test output

```
    Finished `test` profile [unoptimized + debuginfo] target(s) in 50.25s
     Running unittests src/main.rs (target/debug/deps/part1-4688eef695015c56)

running 3 tests
test algorithms::test_build_collaboration_graph ... ok
test algorithms::test_find_connected_components ... ok
test algorithms::test_shared_attributes ... ok

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

Tests Explanation:

algorithms.rs

test_build_collaboration_graph: Verifies that the collaboration graph is built as expected.
test_find_connected_components: Checks that clusters are correctly identified in a simple graph.
test_find_shared_attributes: Ensures shared attributes are detected between freelancers.

Part 2:

Cargo test output

```
running 1 test
test regression::test_basic_regression ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Tests Explanation:

regression.rs

test_basic_regression: Verifies regression works on a simple, linearly related dataset.

E. Results

Part 1 Output:

```
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 1.38s
     Running `target/debug/part1`
Cluster 1 Analysis:
- Members: 256
- Average Earnings: $4888.15
- Average Hourly Rate: $51.50

Cluster 2 Analysis:
- Members: 248
- Average Earnings: $5201.45
- Average Hourly Rate: $50.45

Cluster 3 Analysis:
- Members: 238
- Average Earnings: $5081.07
- Average Hourly Rate: $50.35

Cluster 4 Analysis:
- Members: 231
- Average Earnings: $5094.26
- Average Hourly Rate: $54.09

Cluster 5 Analysis:
- Members: 244
- Average Earnings: $5135.54
- Average Hourly Rate: $54.14

Cluster 6 Analysis:
- Members: 231
- Average Earnings: $4909.05
- Average Hourly Rate: $54.68

Cluster 7 Analysis:
- Members: 265
- Average Earnings: $5136.87
- Average Hourly Rate: $51.48

Cluster 8 Analysis:
- Members: 237
- Average Earnings: $4677.33
- Average Hourly Rate: $54.31
```

This is the output from the analyze_cluster_performance() function. It shows that Cluster 2 has the highest Average Earnings, and Cluster 6 has the highest Average Hourly Rate. We can use this information to conduct further analysis in the next step.

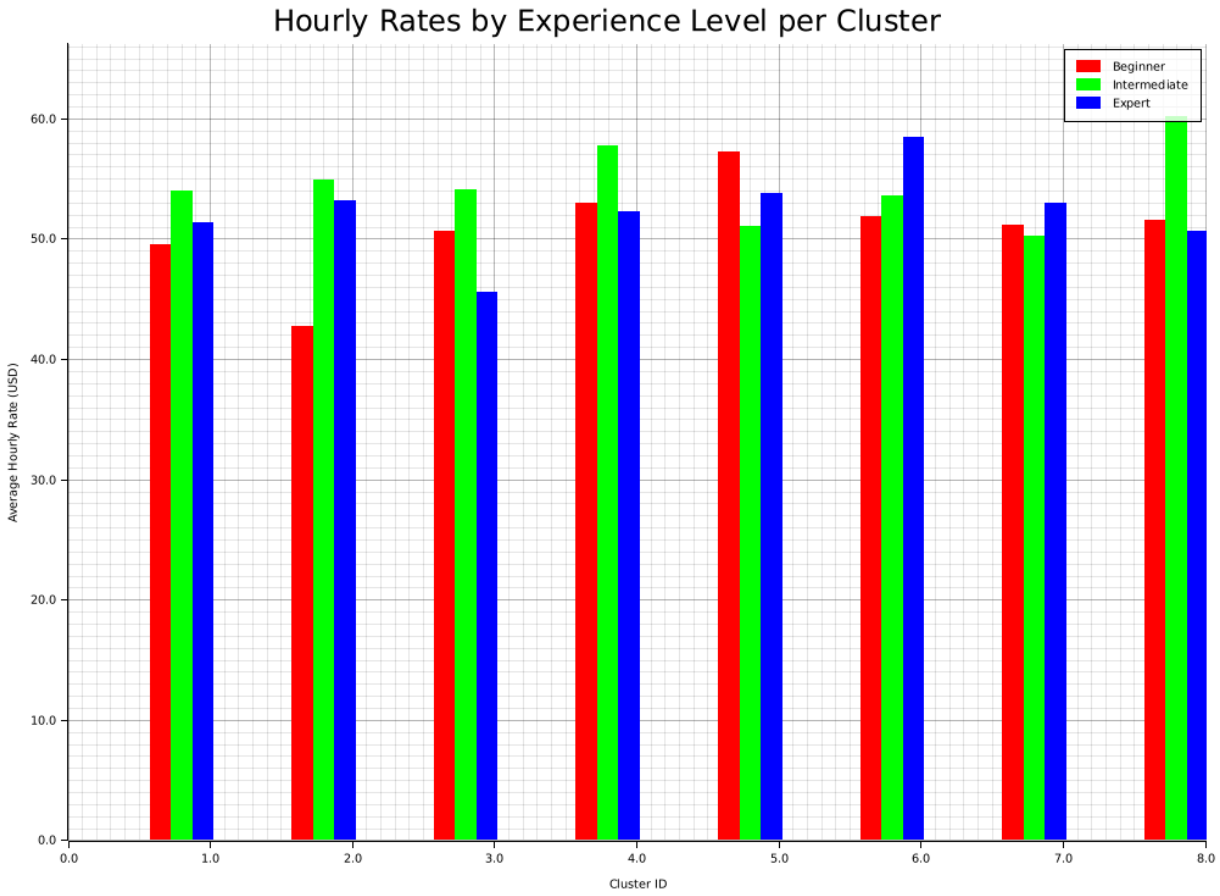
```
Cluster 1 Profile (256 members):
- Dominant Job Category: Web Development (100.0%)
- Dominant Platform: Fiverr (24.6%)
- Dominant Region: UK (19.5%)
- Dominant Experience: Beginner (36.7%)

Cluster 2 Profile (248 members):
- Dominant Job Category: App Development (100.0%)
- Dominant Platform: Toptal (22.2%)
- Dominant Region: Australia (22.2%)
- Dominant Experience: Intermediate (36.3%)

Cluster 3 Profile (238 members):
- Dominant Job Category: Data Entry (100.0%)
- Dominant Platform: Freelancer (21.0%)
- Dominant Region: Australia (16.8%)
- Dominant Experience: Beginner (36.6%)

Cluster 4 Profile (231 members):
- Dominant Job Category: Digital Marketing (100.0%)
- Dominant Platform: Freelancer (22.9%)
- Dominant Region: USA (17.7%)
- Dominant Experience: Beginner (37.7%)

Cluster 5 Profile (244 members):
- Dominant Job Category: Customer Support (100.0%)
- Dominant Platform: Upwork (25.4%)
- Dominant Region: Asia (17.2%)
- Dominant Experience: Expert (37.3%)

Cluster 6 Profile (231 members):
- Dominant Job Category: Content Writing (100.0%)
- Dominant Platform: Upwork (21.2%)
- Dominant Region: Europe (16.5%)
- Dominant Experience: Intermediate (36.4%)

Cluster 7 Profile (265 members):
- Dominant Job Category: Graphic Design (100.0%)
- Dominant Platform: Upwork (22.3%)
- Dominant Region: Australia (17.0%)
- Dominant Experience: Beginner (33.6%)

Cluster 8 Profile (237 members):
- Dominant Job Category: SEO (100.0%)
- Dominant Platform: Upwork (25.7%)
- Dominant Region: Middle East (16.0%)
- Dominant Experience: Intermediate (35.0%)
```

From the last step, we want to take a closer look at Cluster 2 and 6. We can see that all the members in a Cluster have the same Job Category, since we put a lot of weight on this determinant in grouping the datapoints.

The Job Category for Cluster 2 is App Development. From the last step, we can see that this Cluster has a fairly low average hourly rate compared to other Clusters, even though it has the average earnings. Therefore, we can conclude that Freelancers working in App Development require long working hours, but have pretty good job opportunities.

The Job Category for Cluster 6 is Content Writing. From the last step, we can see that this Cluster has the highest average hourly rate, and medium average earnings. We can then draw the conclusion that Freelancers working in Content Writing have a relatively easy workload with a decent payoff compared to others.

Plot:



The generated cluster_experience_rates.png shows bar groups for each cluster, with bars for Beginner, Intermediate, and Expert hourly rates, color-coded and with a legend.
Interpretation:

From the graph, we can see that Cluster 5(Customer Support) is the most beginner-friendly, and Cluster 6(Content Writing) has the best payoff for experts in the field.

Part 2 Output:

```
    Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.92s
     Running `target/debug/part2`
Model Results:
Intercept: 51.48

Coefficients:
Job Success Rate (0-1): 0.21
Job Category (1-5): -1.20
Experience Level (1-3): 0.67

Example Predictions:
Expert Web Developer: $[52.49]/hr
Entry Level Designer: $[48.70]/hr
```

The regression from our model is:
Hourly Rate = 51.58 + 0.21(Job Success Rate) - 1.20(Job Category) + 0.67(Experience Level)

We can see that Job Category is a strong factor for the hourly rate.

F. Usage Instructions

This project doesn't have any user interaction; therefore simply use (Cargo run) or (Cargo run –release) to run the code.
The expected run time is under 30 seconds for both Part 1 and Part 2, with Part 1 taking a bit longer to compile.

G. AI-Assistance Disclosure and Other Citations

I consulted GPT-4 on the graphing part of the code. I asked about how to set colors for the bars, customize the legends, and adjust the positioning of the bars to align with the axis.
The response from GPT-4 is the explanation of the way to do those with sample code. I followed the method provided to complete my plotting function for Part 1 in the analysis.rs module.