Day 6: Multiple Linear Regression: Predicting House Prices

☆ **Points: 390.30000000000007 Rank: 1302**

Problem Submissions Leaderboard Editorial

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Objective

In this challenge, we practice using multiple linear regression to predict housing prices. Check out the Resources tab for helpful videos!

Task

Charlie wants to buy a house. He does a detailed survey of the area where he wants to live, in which he quantifies, normalizes, and maps the desirable features of houses to values on a scale of 0 to 1 so the data can be assembled into a table. If Charlie noted F features, each row contains F space-separated values followed by the house price in dollars per square foot (making for a total of $F+1$ columns). If Charlie makes observations about H houses, his observation table has H rows. This means that the table has a total of $(F+1) \times H$ entries.

Unfortunately, he was only able to get the price per square foot for certain houses and thus needs your help estimating the prices of the rest! Given the feature and pricing data for a set of houses, help Charlie estimate the price per square foot of the houses for which he has compiled feature data but no pricing.

Important Observation: The prices per square foot form an approximately linear function for the features quantified in Charlie's table. For the purposes of prediction, you need to figure out this linear function.

Recommended Technique: Use a regression-based technique. At this point, you are not expected to account for bias and variance trade-offs.

Input Format

The first line contains 2 space-separated integers, F (the number of observed features) and N (the number of rows/houses for which Charlie has noted both the features and price per square foot).
The N subsequent lines each contain $F+1$ space-separated floating-point numbers describing a row in the table; the first F elements are the noted features for a house, and the very last element is its price per square foot.

The next line (following the table) contains a single integer, T, denoting the number of houses for for which Charlie noted features but does not know the price per square foot.
The T subsequent lines each contain F space-separated floating-point numbers describing the features of a house for which pricing is not known.

Constraints

- $1 \leq F \leq 10$
- $5 \leq N \leq 100$
- $1 \leq T \leq 100$
- $0 < \text{ Price Per Square Foot } < 10^6$

- $0 \leq$ Factor Values ≤ 1

Scoring

For each test case, we will compute the following:

- $d =$ Normalized Distance from Expected answer $= \frac{abs(Computed - Expected)}{Expected}$

There are multiple ways to approach this problem that account for bias, variance, various subjective factors, and "noise". We take a realistic approach to scoring and permit up to a $\pm 10\%$ swing of our expected answer.

- $d_{adjusted} = max(d - 0.1, 0)$
- Score for each test case $\equiv max(1 - d_{adjusted}, 0)$
- Score for the test case $\equiv$ (Average score for all the tests it contains) $\times M$, where M is the maximum possible score for the test case.

Consider a test case in which we only need to find the pricing for 1 house. Suppose our expected answer is 10, and your answer is 9.5:

$$d = \frac{(10-9.5)}{10} = 0.05$$

$$d_{adjusted} = max(0.05 - 0.1, 0) = 0$$

The score for a test case with 10 points $= max(1, 0) \times 10 = 10$

Output Format

Print T lines, where each line i contains the predicted price for the i^{th} house (from the second table of houses with unknown prices per square foot).

Sample Input

```
2 7
0.18 0.89 109.85
1.0 0.26 155.72
0.92 0.11 137.66
0.07 0.37 76.17
0.85 0.16 139.75
0.99 0.41 162.6
0.87 0.47 151.77
4
0.49 0.18
0.57 0.83
0.56 0.64
0.76 0.18
```

Sample Output

```
105.22
142.68
132.94
129.71
```

Change Theme C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input Run Code Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature