Table: `Insurance`

```
+-------------+-------+
| Column Name | Type  |
+-------------+-------+
| pid         | int   |
| tiv_2015    | float |
| tiv_2016    | float |
| lat         | float |
| lon         | float |
+-------------+-------+
pid is the primary key column for this table.
Each row of this table contains information about one policy where:
pid is the policyholder's policy ID.
tiv_2015 is the total investment value in 2015 and tiv_2016 is the total investment value in
lat is the latitude of the policy holder's city. It's guaranteed that lat is not NU
lon is the longitude of the policy holder's city. It's guaranteed that lon is not NU
```

Write an SQL query to report the sum of all total investment values in 2016
`tiv_2016`, for all policyholders who:

have the same `tiv_2015` value as one or more other policyholders, and

are not located in the same city like any other policyholder (i.e., the (`lat`, `lon`)
attribute pairs must be unique).

Round `tiv_2016` to **two decimal places**.

The query result format is in the following example.

Example 1:**

```
Input:
Insurance table:
+-----+----------+----------+-----+-----+
| pid | tiv_2015 | tiv_2016 | lat | lon |
+-----+----------+----------+-----+-----+
| 1   | 10       | 5        | 10  | 10  |
| 2   | 20       | 20       | 20  | 20  |
| 3   | 10       | 30       | 20  | 20  |
| 4   | 10       | 40       | 40  | 40  |
+-----+----------+----------+-----+-----+
Output:
+----------+
| tiv_2016 |
```

```
+----------+
| 45.00    |
+----------+
```
Explanation:
The first record in the table, like the last record, meets both of the two criteria.
The tiv_2015 value 10 is the same as the third and fourth records, and its location is uniqu

The second record does not meet any of the two criteria. Its tiv_2015 is not like any other
So, the result is the sum of tiv_2016 of the first and last record, which is 45.