Store planner software

Back story

Your father owns a little business and he wants you to help him improve his postions. As you are a freshman and you are still not good at applying Machine Learning models, you can only help him find the closest store with one parametr as an input. Your father have a .csv file with list of his competitors' stores. He wants to input one of stores parametrs to the app and get the profit he can get with this parametr in his mind. Help your father become da best business boi out there!

Which functionality shoud your app support

1. Read .csv files
2. Show .csv files
3. Add/delete rows and columns
4. Properly handle blank cells
5. Export edited .csv files
6. Output possible income based on store parametr

Given data

You are given a datasets to deal with. It is "Supermarket store branches sales analysis", it has multiple columns representing different characteristic of a store and sales of this shop.

- Store ID - technical parametr
- Store Area - feature parametr
- Items available - feature parametr
- Daily customers - feature parametr
- Store sales - target parametr (actually can also be a feature parametr)

[Store Sales Dataset](#)

This data set is very well prepared, it doesn't have any blank cells, data is highly correlated.

If something accidentally happens to these links and datasets on Kaggle.com, you can contact me through Sergey Shershakov as I have them backed up on my local machine and

cloud storage

Working with .csv file

I would like you to add some additional stuff to edit given .csv file:

- Add rows
- Delete rows
- Implement analog of fillna method from python package "Pandas" but using some magic: you should not just fill the cell with the data from upper or lower row, but try finding the closest value to this one by other left store characteristics (the importance of values is Area -> Customers -> Sales -> Items) so, for example if you are lacking the Sales cell, you first look for 50 closest stores on Area, after that you look for 25 closest on Customers value (across 50 closest on Area) and so on. This numbers(50, 25) you can choose by yourself. I hope the main idea is clear.

User interface design

I would like to mention that I kindly ask you to stick to this UI design as your father is not a tech-wizard and he needs it to be as simple as it could posibly be. This is how the main window should look like.
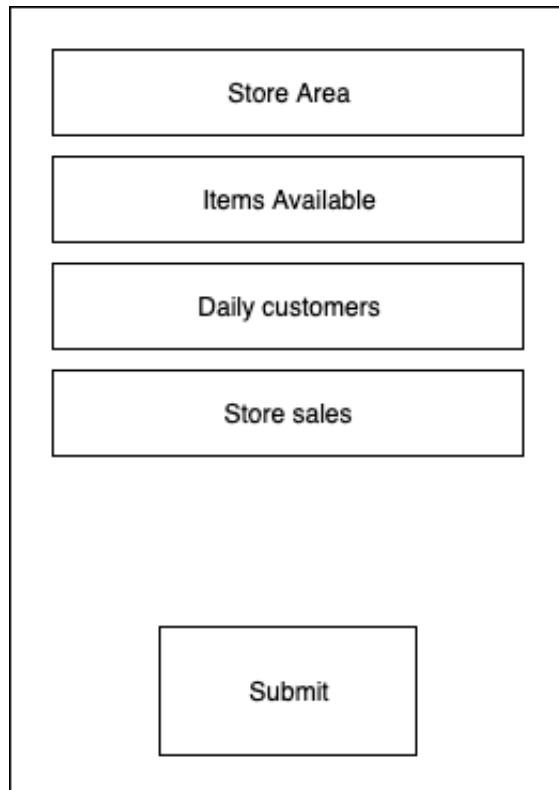

Now lets talk about what each button here does:

1) Button "Import csv" opens your Finder if you are a mac user, and Explorer if you are a windows users, you choose your file (given dataset) and after pressing enter - it shows up on the spot called csv table.

2) Button "Save csv" opens your file explorer as well, and you perform usual operation of saving a file as you have done many times before in your life.

3) Button "Add store" opens up a dialog window, which looks like this:

Store Area

Items Available

Daily customers

Store sales

Submit

here you should be able to fill in the gaps for at least 3/4 lines. The last line, if left blank - should be filled using your custom "fillna" method described above.

4) Button "Delete store" opens up a dialog window, which looks like this:

Store Id

Delete

If you wish, you can add some kind of confirmation of "delete" operation.

5) Button "About" opens up a dialog window, which looks like this:

Project "Papa"

Author - StudenXXX

Distribution license CC NC

```
        close
```

You can write whatever info you wish to, but there should at least be your "Student XXX".

6) Button "Edit store" opens up a dialoge window, which looks like this:

```
    Store Id

    load
```

After entering store id you and pressing load button - you will get to the next dialog window:

```
    Store area - value for id

    Items Available - value for id

    Daily customers - value for id

    Store sales - value for id




        Submit
```

Where you are shown all actual valuse for the idth store and where you can change each of them.

7) Button "Fill blank" automatically calls your "fillna" method for all blank cells.

8) Button "Find closest settings" is kind of optional. It is all about the importance of cells for your "Find Closest method" and for which value you are going to find it. For simplicity you can skip this and just make some pre-defined importance, but if you decide to implement it - it opens up a dialogue window:

Each param box is a drop-down list of 4 params (your table columns without ID column). You should put them in order of importance with the most important on top. After clicking "update" params are saved and dialogue windows closes.

9) Button "Find Closest" opens dialogue window which looks like this:

Here you choose the parametr(feature which you want to guess from) from a drop down list and the target param choosed the same way, after that you enter the feature value and push "predict button", after that Corresponding target value box should be filled with value. The mechanism for finding the closest is the same as Filling blank cells but with minor changes. Suppose our we have a value X of column Store Area. So we need to find such ID so that X - x(id)(value of the same column as X for store "id") is the least possible and output target parametr of id-th store.

That is it! Good luck!