Task 1 Group 6 README

Jack Cook

Govardhan Digumurthi

Thomas Okonkwo

Description of dataset

The spark program is used to determine the number of features (columns) and rows that are in the dataset. The schema of the dataset is also displayed. The schema is an overview of the column names and the variable types that are in each column (or feature).

At the very bottom of the data set (row 11,583) there exists a description of what some of the keys in the dataset mean:

- Suburb - name of suburb
- Type:
  - h - house, cottage, villa, semi, terrace
  - u - unit, duplex
  - t - townhouse site, development side
  - o - res, other residential
- Price: Price in dollars ($)
- Distance: Distance from CBD
- Zipcode: the zipcode where the unit is located
- Bedroom: number of bedrooms
- Bathroom: number of bathrooms
- Car Garage: number of car garages
- Lot Size: the size of the property (in acres)
- Region name: General region (West, North West, North, North East, etc.)
- Property count: Number of properties that exist in the suburb

There are 7 region names, 3 housing types and 312 suburbs.

Work done

There are rows at the bottom at the data that helped to describe the data, but it did not contain any meaningful data for processing and has been removed.

Here is a list of the csv files and a description of what they are:

- Housing_data-Final-1.csv - the original csv supplied
- Housing_data-Final-2.csv - the bottom 20 rows of garbage data are removed

The following is a list of tasks performed in the code in order: a spark configuration is defined, a csv file is read in from a command line argument, the dataframe is analyzed, the number of rows in the columns are found, the schema for the dataframe is printed to the console so that the types in each column can be understood, the columns of type string are analyzed to see if there is any foul data, the string values ae replaced with numeric values, the dataframe is split into a good and bad dataframe, cosine similarity is used to replace the bad cells with similar values from good rows (normalization is necessary for making cosine similarity work properly), the fixed dataframe is merged into the good dataframe and the results are ouptut to `Group6_Task1_Output.csv`.

The following examples are at the bottom of the file and are listed in order: a simple cosine similarity using dense user defined spark vectors to understand the function, computation of cosine similarity of the first row to all other rows in a dataframe, normalization of the data and then computation of cosine similarity of the first row versus all other rows.

Instructions to run program

You must be logged into the hadoop cluster using `ssh cookjc@hadoop-nn001.cs.okstate.edu`, with the username in this example being `cookjc`.

The housing csv file that has had the bottom 20 garbage rows removed needs to be stored in the hadoop file system:

```
hdfs dfs -copyFromLocal Housing_data-Final-2.csv /user/cookjc
```

The file must be executed using `spark-submit`:

```
spark-submit Group6_Task_1_Code.py Housing_data-Final-2.csv
```

Discussion of results

The bad cells in the data are corrected by cells in the same column in rows that are similar. The similarity is determined by first converting all striings to integers, normalizing the data and then computing similarity based on the columns that do not contain bad data.

The combiend dataframe has 11,579 rows, just like the original dataset had.