🥚 CS 210 Homework 3 🐤

Fall 2022

Overview

In this assignment, you're asked to do analysis and visualization of three datasets:

- EU city temperatures
- German credit
- Google playstore apps

Please write your code in Jupyter notebooks named `eucities_temps.ipynb`, `german_credit.ipynb`, and `google_apps.ipynb`.

In each notebook, please add markdown cells where appropriate so it's clear what part of the problem you're solving.

Zip all of these notebooks into a file named `hw3.zip` and submit this on Canvas. Do not include any input or output files.

You may only import modules we have covered in lecture, such as numpy, pandas, and matplotlib.

🍳 Problem 1: EU Cities Temperatures (30 points)

Load the CSV file `EuCitiesTemperatures.csv` (213 records) into a Pandas DataFrame and perform the following tasks on it.

Preprocessing/Analysis (15 points)

1. (5 points) Fill in the missing latitude and longitude values by calculating the average for that country. Round the average to 2 decimal places.

2. (5 points) Find the subset of cities that lie between latitudes 40 to 60 (both inclusive) and longitudes 15 to 30 (both inclusive). Find out which countries have the maximal number of cities in this geographical band. (More than one country could have the maximal number of values.)

3. (5 points) Fill in the missing temperature values by the average temperature value of the similar region type. A region type would be a combinaton of whether it is in EU and whether it has a coastline.

   For example, if we have a missing temperature value for Bergen, Norway, which is not in the EU but lies on the coast, we will fill it with the average temperature of cities with `EU='no'` and `coastline='yes'`.

Visualization (15 points)

For all plots, make sure to label the axes, and set appropriate tick labels.

1. (3 points) Plot a bar chart for the number of cities belonging to each of the regions described in Preprocessing/Analysis #3 above.

2. (3 points) Plot a scatter plot of latitude (y-axis) vs. longitude (x-axis) values to get a map-like visual of the cities under consideration. All the cities in the same country should have the same color.

3. (4 points) The population column contains values unique to each country. So two cities of the same country will show the same population value. Plot a histogram of the number of countries belonging to each population group: split the population values into 5 bins (groups).

4. (5 points) Create subplots in a 2×2 grid, with proper titles, one each for the region types described in Preprocessing/Analysis #3 above.

Each subplot should be a scatter plot of Latitude (y-axis) vs. City (x-axis), where the color of the plot points should be based on the temperature values: red for temperatures above 10, blue for temperatures below 6, and orange for temperatures between 6 and 10 (both inclusive).

For each subplot, set xticks to an array of numbers from 0 to $n-1$ (both inclusive), where n is the total number of cities in each region type.

🍳 Problem 2: German Credit (35 points)

Load the CSV file `GermanCredit.csv` (1000 records) into a Pandas DataFrame, and perform the following tasks on it.

Preprocessing (10 points)

1. (2 points) Drop the 3 columns that contribute the least to the dataset. These would be the columns with the highest number of non-zero 'none' values. Break ties by going left to right in columns. (Your code should be generalizable to drop n columns, but for the rest of the analysis, you can call your code for $n = 3$.)

2. (2 points) Certain values in some of the columns contain unnecessary apostrophes (`'`). Remove the apostrophes.

3. (2 points) The `checking_status` column has values in 4 categories: `'no checking'`, `'<0'`, `'0<=X<200'`, and `'>=200'`. Change these to `'No Checking'`, `'Low'`, `'Medium'`, and `'High'` respectively.

4. (2 points) The `savings_status` column has values in 4 categories: `'no known savings'`, `'<100'`, `'100<=X<500'`, `'500<=X<1000'`, and `'>=1000'`. Change these to `'No Savings'`, `'Low'`, `'Medium'`, `'High'`, and `'High'` respectively. (Note that the last two are both `'High'`).

5. (2 points) Change class column values from `'good'` to `'1'` and `'bad'` to `'0'`, and change the employment column value `'unemployed'` to `'Unemployed'`, and for the others, change to `'Amateur'`, `'Professional'`, `'Experienced'` and `'Expert'`, depending on year range.

Analysis (10 points)

For the following tasks, do preprocessing or changing of data types in the data frame as required.

1. (4 points) Often we need to find correlations between categorical attributes, i.e. attributes that have values that fall in one of several categories, such as "yes"/"no" for attr1, or "low","medium","high" for attr2.

   One such correlation is to find counts in combinations of categorial values across attributes, as in how many instances are "yes" for attr1

and "low" for attr2. A good way to find such counts is to use the Pandas `crosstab` function. Do this for the following two counts.

   (a) (2 points) Get the count of each category of foreign workers (yes and no) for each class of credit (good and bad).

   (b) (2 points) Similarly, get the count of each category of employment for each category of `saving_status`.

2. (2 points) Find the average `credit_amount` of single males that have $4 \leq x < 7$ years of employment. You can leave the raw result as is, no need for rounding.

3. (2 points) Find the average credit duration for each of the job types. You can leave the raw result as is, no need for rounding.

4. (2 points) For the purpose `'education'`, what is the most common `checking_status` and `savings_status`? Your code should print:

```
Most common checking status: ...
Most common savings status: ...
```

Visualization (15 points)

1. (5 points) Plot subplots of two histograms: one with `savings_status` on the x-axis and `personal_status` as different colors, and another with `checking_status` on the x-axis and `personal_status` as different colors.

2. (5 points) For people having `credit_amount` more than 4000, plot a bar graph which maps `property_magnitude` (x-axis) to the average customer age for that magnitude (y-axis).

3. (5 points) For people with a "High" `savings_status` and age above 40, use subplots to plot the following pie charts:

   (a) Personal status

   (b) Credit history

   (c) Job

🔍 Problem 3: Google Playstore Apps (35 points)

Load the Excel data file `GooglePlaystore.xlsx` (10K records) into a Pandas DataFrame (see the Pandas `read_excel` method), and perform the following tasks on it.

Preprocessing (15 points)

1. (2 points) Often there are outliers which do not match the overall data type. There is one record in this data where the "Reviews" has value "3.0M" which does not match the rest of the data. Remove that record.

2. (2 points) Remove rows where any of the columns has the value "Varies with device".

3. (2 points) The values in the Android version column should be floats. Strip the trailing non-numeric characters from all values (ie. the words " and up"), so the result is a number. If there are multiple decimal places (eg. "x.y.z"), keep only the first two parts (eg "x.y"). For example, the value "4.1 and up" should be changed to "4.1". The value "4.5.6 and up" should be changed to "4.5". The value "5.6.7" should be changed to "5.6".

   If there is a range (eg. 5.0 – 8.0), only consider the first number. For example, the value "5.0 – 8.0" should be changed to "5.0". The value "4.0.3 – 7.1.1" should be changed to "4.0".

4. (3 points) The "Installs" column must have integer values. For values that have commas, remove the commas. For values that have a '+' at the end, remove the '+'. Keep only those rows that have an integer value after these edits.

5. (3 points) For missing rating values, if the number of reviews is less than 100 and installations is less than 50000, remove the row. Else, fill the missing value with the average value (rounded to 2 decimal places) for the Category of that row.

6. (3 points) In Size column, convert "M" (millions) and "K" (thousands) values into integers. For instance, 8.7M should be converted to 8700000 and 2.4K should be converted to 2400.

Analysis (10 points)

For the following tasks, do preprocessing or changing of data types in the data frame as required.

1. (2 points) Describe (use DataFrame `describe` method) the category-wise rating statistics. In other words, for each category, describe the statistics (count, mean, etc.) for ratings in that category.

2. (5 points) Extract all "Free" apps from the master data frame. Then write a function that, given a numeric column e.g, `'Rating'`, will create and return a dataframe for the top 3 free applications in each category based on that column. Call the function on each of these columns:

   (a) Rating (gives 3 most highly rated applications in each category)

   (b) Installs (gives 3 most installed applications in each category)

   (c) Reviews (gives 3 most reviewed applications in each category)

   You don't need to do anything explicit to break ties.

   Each of the returned dataframes have Category and App for the first two columns, and one of Rating (for (a)), Installs (for (b)), and Reviews (for (c)) as the third column.

3. (3 points) Find the average, maximum and minimum price of the paid applications.

Visualization (10 points)

1. (5 points) In the genre column, break the string of genres into a list. For example, 'Art & Design; Creativity' should be ['Art & Design', 'Creativity'].

   Count the number of applications per genre and display it using a pie chart.

   Hint: See `DataFrame.explode()`

2. (5 points) Display a box plot of ratings for "Business" and "Education" categories. The boxplots should be in the same plot.