Pre-DSI Assignment - 2022

Welcome to the Pre-DSI assignment!

Please read these instructions carefully.

The main goal of this assignment is to help you make sure you have got the basic tools needed for you to be successful in the intense, highly pressurised environment of the DSI. It is derived from feedback from DSI alumni who wished that they had had this assignment before starting...

You already have a conditional acceptance to the DSI. The key point is that if you struggle with any element of this assignment it is a good indicator that you need to get up to speed in the corresponding area **before** the DSI. Please don't ignore this, the DSI is not the time to start learning core skills that you will need!!

We trust that 6 weeks is enough time to get the basics you will need, but if we are still concerned that you may not have the minimum skills needed to contribute sufficiently to your team in the DSI we will ask you to do a live, in-depth coding interview. If we are still concerned that your python or other basic data science skills are not sufficient to ensure your success on the DSI then we will have to withdraw the offer of a place on the DSI.

Past experience has shown that people who are equipped with the basics do not enjoy the DSI and do not succeed.

If you choose not to submit the assignment (unless you make alternative arrangements with us **before** 1 Jan 2022) we will assume you are withdrawing from the DSI and offer your place to the next person on the very long waiting list.

This assignment will take you through some of the key python and pandas skills you will need to succeed at the DSI. Our advice is to complete this assignment as soon as you can for a simple reason: if you find that you struggle you want to give yourself as much time to learn the basics that are missing as you can!

Good luck and we hope you enjoy this - it is meant to be fun *and* challenging!

If you have any questions please contact nadeem@dsi-program.com, the academic director of the DSI.

Submission Deadline: 10 Jan 2022

Live Coding Interviews: 13-18 Jan 2022

Question 1: Unexpected Plots

Write a python function that will plot the points that come from the following procedure:

1. Create a **regular** hexagon
2. Pick a *random* point, P, inside the hexagon.
3. Make a triangle, T, by *randomly* connecting P to two *adjacent* vertices of the hexagon.
4. Compute the centroid of T. This becomes your new random point, P. Save it, make a new random triangle as above, compute the new centroid etc… Repeat this process 10,000 times.
5. Make a scatter plot of all your 10,000 random points. What emerges?

The goal of this exercise is to ensure that you have enough basic skills and familiarity with python to succeed.

Question 2: Datascience with Pandas and Movie Data

Pandas, jupyter notebooks and colab are some of the most basic python tools in data science. Here we want to make sure you know how to load data into pandas, manipulate columns, rows and subsets of dataframes, perform string manipulation, regex, merging of dataframes, groupby, sorting of values, as well as know how to perform basic statistics on dataframes.

We also want you to be familiar with installing python packages, reading documentation and have a little familiarity with key libraries like scipy, sklearn and pytorch.

Finally we also want you to show that you understand the basics of git, which is an industry standard.

Instructions:

Each of these questions should be answered in a separate cell in your jupyter notebook.

0. Access some historical IMDB data files from the shared drive:

https://drive.google.com/drive/folders/1dl6nw0HO9XVrT8dSBJHHn3mDW9EWQpXS?usp=sharing

1. Read the files 'title.basics.tsv.gz', 'title.akas.tsv.gz' and 'title.ratings.tsv.gz' into three separate dataframes using the read_csv method in Pandas.

2. Drop duplicates in all the dataframes, if there are any.

3. Using the Pandas 'merge' method, combine all three dataframes using the Title ID (titleID or tconst) to perform the merge and save it into a new dataframe.

   3.1 How many lines does the resulting dataframe have if you use an inner merge or outer merge? Make sure you understand the difference.

   3.2 Using the unique() method, compute how many different 'titleTypes' there are

4. Make a new dataframe from step 3 by selecting only rows corresponding to English-language films ('en') OR US-region films ('US') AND only those that are movies (using the 'titleType' column). Put the resulting data into a new dataframe; call it df_new.

5. Add a new column to df_new with column title 'log10Votes' which gives the Log_10 number of the 'numVotes' column.

6. Lower the case of all text in the 'genres' column.

6. Using Groupby (or other technique) group all data by 'genres' and display the top 10 highest genres by:

   6.1 mean number of log10Votes
   6.2 mean averageRating

7. Using ‘groupby’ group all data by averageRating and make a scatter plot of averageRating vs log10Votes.

8. Perform linear regression on your data (averageRating vs log10Votes) created in the previous step in three different ways:

   8.1 Using sklearn
   8.2 Using scipy
   8.3 Using pytorch

Ensure that you get the same result in each case (or explain why the results are different). You will need to install the corresponding packages. If you wanted to build a better regression model what would you do?

9. You should commit at least three different versions of your notebook to your github account to demonstrate that you know the basics of using git for version control.

10. Share your notebook with us as a Google Colab notebook. **NB: MAKE SURE TO MAKE IT PUBLIC**. Include your github account in your Colab notebook intro and make sure your commits are public.

Resources

If you struggle with any of these questions there are a large number of excellent resources to help you that you can find by Googling around, but here are some to get you started:

- Datacamp.com
- https://www.codecademy.com/learn/data-processing-pandas/modules/dspath-intro-pandas
- https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html
- https://realpython.com/pandas-groupby/ (for groupby)