Week 1 - Data Exploration

TLDR

[Sign up](https://docs.google.com/spreadsheets/d/1XUcYdy1TeWcH0T7XWIma8ij3ctAcNZ7zIKvKahJOiyk/edit?usp=sharing) by **Friday September 28** to present next week or if you would like someone to present on your behalf. Everyone must present by the end of the semester!

By **Wednesday October 3**, regardless of whether you are presenting or not, please complete the steps listed in the Getting Started and Adding Your Exploration Code sections below.

Overview

[Project Meeting Slides](https://docs.google.com/presentation/d/14Vn3IjGvxERqs2DJhwI-56vll2xwDkmw3Z6YSyPryzg/edit?usp=sharing)

[Github](https://github.com/MichiganDataScienceTeam/googleanalytics)

[Slack](https://mdst-chat.slack.com/messages/CCVN3CX89/details/)

This week, we will explore the Google Analytics dataset. By the end of the week, you should be able to load the dataset onto your computer, run some starter code, and understand the basic structure of the data. Each person will be responsible for exploring one aspect of the data in depth, and we will use our knowledge to build a shared exploratory notebook!

Why do data exploration?

* We often discover surprising trends or patterns just by looking at the data.
* Many datasets have missing or not “clean” data, which are easily overlooked.
* The patterns we find can be used to build better predictive models.

Data exploration should always be the first thing you do when you start a new project.

The Dataset

“In this competition, you’re challenged to analyze a Google Merchandise Store (also known as GStore, where Google swag is sold) customer dataset to predict revenue per customer. Hopefully, the outcome will be more actionable operational changes and a better use of marketing budgets for those companies who choose to use data analysis on top of GA data.”

The dataset is hosted on Kaggle as part of the Google Analytics Customer Revenue Prediction Challenge. This will be covered in some detail in the project meeting. For more information, please refer to the [Kaggle competition page](https://www.kaggle.com/c/ga-customer-revenue-prediction).

The Competition

We will compete, as a single team, in the [Google Analytics Customer Revenue Prediction Challenge](https://www.kaggle.com/c/ga-customer-revenue-prediction) on Kaggle. By a “single team”, we mean that all MDST members will work together to create a single submission to this competition! Please note that, in order to comply with Kaggle’s rules on maximum team size, we must develop all of our code publicly. Because of this, **please do not share code or data privately with anyone else on MDST**, and **please do not submit any results to Kaggle**. More about this will be discussed in our project meeting.

The prediction target is the [natural log of the sum of all transactions per user](https://www.kaggle.com/c/ga-customer-revenue-prediction/data). This is the variable that we will predict and submit to Kaggle.

The evaluation metric is [Root Mean Squared Error (RMSE)](https://www.kaggle.com/c/ga-customer-revenue-prediction#evaluation). The Kaggle leaderboard is ordered using this metric, which is computed on a fraction of the test data. The test labels are never revealed to us, the competitors.

Starter Code

We have created a [Github repo](https://github.com/MichiganDataScienceTeam/googleanalytics) where we will develop all of our code as a team. At the moment, the code only loads the dataset from a file and converts it into the [Dataframe](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html) format. Here are the important files you need to know for this week:

data/ this is the directory where you store the competition data.

train.py does nothing for now, please ignore

dataset.py defines how to load the dataset

explore\_utils.py defines several functions that explore the dataset

explore.py calls all of the functions from explore\_utils.py

Right now, you should be able to run python explore.py, which will print out some basic info about the dataset. Before moving on, please make sure this works.

Getting Started

Before you contribute to the repository, please complete the following steps.

1. Join the #google-analytics slack channel.
2. Make an account on Github if you don’t have one already
3. Message @stroud with your Github account name to join the MDST Github organization.
   1. Note: you can start step 3 before @stroud gets back to you.
4. Clone the data, run the code, and make sure it works.
   1. Do not download as a .zip. Please follow the instructions in the README.
5. If you run into any issues with step 3, please create a Github issue in the repo.

Do not hesitate to reach out for help if you are confused about any of the above steps. Keep in mind that most MDST members are not familiar with Git or Github when they join. You may post issues on Github, post in the #google-analytics Slack channel, or reach out to @stroud on Slack.

Adding Your Exploration Code

This week, you will modify **explore\_utils.py** and **explore.py**. You will write one function which explores some aspect of the Google Analytics dataset. At the end of the week, we will aggregate everyone’s contributions into a single notebook!

**Step 0.** We will have six MDST members present their results each week. If you are able to present the results of your exploration in next week’s project meeting, or you would like someone to present on your behalf, please fill out the [**sign up sheet**](https://docs.google.com/spreadsheets/d/1XUcYdy1TeWcH0T7XWIma8ij3ctAcNZ7zIKvKahJOiyk/edit?usp=sharing) by Friday, September 28th. More details will be shared in the project meeting this week.

**Step 1.** Look at the [Github Issues](https://github.com/MichiganDataScienceTeam/googleanalytics/issues) tagged with “exploration”. Find one that you want to work on, and click “assign yourself”. You will need to be a member of the MDST Github organization to complete this step. Please only select issues that haven’t already been chosen. You may also create a new issue.

**Step 2**. Create a new branch. Give it a descriptive name which is related to the issue you have chosen. You can create a new branch and switch to it like this:

git checkout -b [name\_of\_your\_new\_branch]

**Step 3**. Add your function to **explore\_utils.py**. Your function should have this form:

def compute\_something\_interesting(dataset):

“””Computes something interesting about the dataset.

args:

dataset (Dataset): the google analytics dataset.

returns:

Something interesting.

“””

train\_df = dataset.train

test\_df = dataset.test

# your code goes here

return something\_interesting

See explore\_utils.py for some examples. **Note**: your function should not modify the dataset in any way. It should only read the dataset. If you need to change the dataset, please create a copy within your function using [pandas.Dataframe.copy](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.copy.html).

**Step 4.** Call your function from **explore.py**. Your call should happen in the main function of explore.py. You should also add some code that prints out the interesting quantity that your function returned. Here is an example:

# Your code starts here

something\_interesting = explore\_utils.compute\_something\_interesting(dataset)

print(“An interesting thing I computed:”, something\_interesting)

If you find that you need to make changes to the code other than what is described here, you should probably do this in a separate branch. Please limit your branches to one change at a time.

**Step 5.** Test your code. Make sure that the code works as expected. We do not have any framework for automatic unit tests at the moment, so please just test your code yourself before you submit. Most importantly, make sure that your code will not have any unintended effects on anyone else’s code. Do not modify the dataset in your function without making a local copy.

**Step 6**. Commit your code. You can do this from the command line as below. Make sure to add a descriptive commit message.

git add explore\_utils.py

git add explore.py

git commit -m “Your descriptive commit message goes here.”

**Step 7**. Push your code to Github. Your new branch should appear on Github after this step.

git push origin name\_of\_your\_new\_branch

**Step 8.** Submit a Pull Request (PR) on Github. You can do this from your browser by clicking the “New Pull Request” button. Please reference the Issue you are working on in your PR description.

**Step 9.** Make changes based on feedback. We will review your code and let you know if there are any problems. You can continue to update and push changes based on the feedback you receive.

**Step 10.** When your code is approved, merge your pull request into the Master branch. Congratulations, you’re done!

**Step 11.** If you are signed up to present this week, please read and complete the presentation steps below.

Creating a Presentation

If you are signed up to present, you will create a 5-minute, 3-slide presentation about the results of your exploration. Please send your slides to @stroud before the meeting.

**Slide 1**. What did you do? Tell us what issue you chose on Github, and how you addressed it. Did you run into any issues? How did you overcome them?

**Slide 2**. What is the result? Give us a single table or chart showing your results. In some cases, the result may be as simple as one number. It does not need to be complicated - in fact, too much complexity will make it difficult for your audience to follow.

**Slide 3**. What does this tell us? Try to interpret your results. Be clear and concise, but try not to make conclusions that are not fully supported by the data. What does the data tell us, and what does it not?

We will have an example presentation in the first project meeting to give you an idea of what a good presentation looks like. If you’re not exactly sure what to do, keep in mind that the whole point of these presentations is for you to practice! We are not expecting every presentation to be perfect.