Data detective challenge!

Fraida Fund

Introduction

In this notebook, we will consider several machine learning tasks, and candidate data sets for them. We will explore the following questions:

- Do these data sets seem appropriate for the task?
- Are there any important limitations of the datasets, or problems that need to be addressed before we use them to train a machine learning model?

In fact, each of these datasets has a significant problem that - if not detected early on - would create a "Garbage In, Garbage Out" situation. See if you can identify the problem with each dataset!

To get you started, I included some code to show you how to read in the data. You can add additional code and text cells to explore the data.

Your work on this challenge won't be submitted or graded. If you think you found the problem with a dataset, share your findings with the class by posting on Ed! (In your post, show evidence from your exploratory data analysis to support your claims.)

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Taxi tip prediction

Scenario

You are developing an app for NYC taxi drivers that will predict what the typical tip would be for a given fare.

You consider using data collected by the NYC Taxi and Limousine Commission on taxi trips. These links are for 2019 data (2020 was probably an atypical year, so we won't use that). Previous years are also available.

- Data link for yellow (Manhattan) taxi trips
- Data link for green (non-Manhattan) taxi trips

Read in data

We'll start by reading in the 2019 Green Taxi trip data. It's a large file and takes a long time to download, so we may interrupt the download in middle (using the Runtime menu in Colab) and just work with the partial data.

In the next couple of cells, `wget` and `wc` are not Python code - they're Linux commands. We can run some basic Linux commands inside our Colab runtime, and it's often helpful to do so. For example, we may use Linux commands to install extra software libraries that are not pre-installed in our runtime, clone a source code repository from Github, or download data from the Internet.

```
!wget "https://data.cityofnewyork.us/api/views/q5mz-t52e/rows.csv?accessType=DOWNLOAD" -O
    2019-Green-Taxi-Trip-Data.csv
```

```
--2021-06-16 09:26:50--
    https://data.cityofnewyork.us/api/views/q5mz-t52e/rows.csv?accessType=DOWNLOAD
Resolving data.cityofnewyork.us (data.cityofnewyork.us)... 52.206.140.199, 52.206.140.205,
    52.206.68.26
Connecting to data.cityofnewyork.us (data.cityofnewyork.us)|52.206.140.199|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘2019-Green-Taxi-Trip-Data.’csv

2019-Green-Taxi-Tri     [  <=>               ] 556.26M  2.15MB/s    in 5m 2s

2021-06-16 09:31:52 (1.84 MB/s) - ‘2019-Green-Taxi-Trip-Data.’csv saved [583280181]
```

Is the cell above taking a long time to run? That's because this data set is very large, and the server from which it is retrieved is not very fast. Since we don't need to explore the whole dataset, necessarily, we can interrupt the partial download by using the Runtime > Interrupt Execution menu option.

Then, we can read in just 10,000 rows of data.

```
df_taxi = pd.read_csv('2019-Green-Taxi-Trip-Data.csv', nrows=10000)
df_taxi.head()
```

```
   VendorID    lpep_pickup_datetime   lpep_dropoff_datetime  \
0       2.0  02/01/2019 12:10:19 AM  02/01/2019 12:21:43 AM
1       2.0  02/01/2019 12:02:16 AM  02/01/2019 12:24:37 AM
2       2.0  02/01/2019 12:37:19 AM  02/01/2019 12:43:07 AM
3       1.0  02/01/2019 12:10:10 AM  02/01/2019 12:12:21 AM
4       1.0  02/01/2019 12:30:19 AM  02/01/2019 12:46:14 AM

  store_and_fwd_flag  RatecodeID  PULocationID  DOLocationID  passenger_count  \
0                  N         1.0            92           135              1.0
1                  N         1.0            66            36              1.0
2                  N         1.0           255           112              1.0
3                  N         1.0            75           238              1.0
4                  N         1.0            75            48              1.0

   trip_distance  fare_amount  extra  mta_tax  tip_amount  tolls_amount  \
0           2.79         11.0    0.5      0.5        3.08          0.0
1           4.46         17.5    0.5      0.5        3.76          0.0
2           1.26          6.0    0.5      0.5        1.46          0.0
3           0.70          4.0    0.5      0.5        0.00          0.0
4           3.90         14.5    0.5      0.5        0.00          0.0

   ehail_fee  improvement_surcharge  total_amount  payment_type  trip_type  \
0        NaN                    0.3         15.38           1.0        1.0
1        NaN                    0.3         22.56           1.0        1.0
2        NaN                    0.3          8.76           1.0        1.0
3        NaN                    0.3          5.30           2.0        1.0
4        NaN                    0.3         15.80           2.0        1.0

   congestion_surcharge
0                   0.0
1                   0.0
2                   0.0
```

```
3                   0.0
4                   0.0
```

Use additional cells as needed to explore this data. Answer the following questions:

- How is the data collected? Is it automatic, or is there human involvement?
- What variable should be the *target variable* for this machine learning problem?
- What variable(s) could potentially be used as *features* to train the model?
- What are our assumptions about the features and the target variable, and the relationships between these? (For example: in NYC, what is a conventional tip amount, as a percent of the total fare? If you are not from NYC, you can find information about this online!) Are any of these assumptions violated in this data?
- Are there variables that should *not* be used as features to train the model, because of potential for data leakage?
- Are there any serious data problems that we need to correct before using the data for this purpose? Explain.

Highway traffic prediction

Scenario

You are working for the state of New York to develop a traffic prediction model for the NYS Thruway. The following Thruway data is available: Number and types of vehicles that entered from each entry point on the Thruway, along with their exit points, at 15 minute intervals.

The link points to the most recent week's worth of available data, but this data is available through 2014.
Link to NYS Thruway data

Read in data

```
url = 'https://data.ny.gov/api/views/4dbf-24u2/rows.csv?accessType=DOWNLOAD&sorting=true'
df_thruway = pd.read_csv(url)
df_thruway.head()
```

```
         Date Entrance Exit  Interval Beginning Time Vehicle Class  \
0  11/03/2020       15   17                        0            2H
1  11/03/2020       15   17                        0            2L
2  11/03/2020       15   17                        0            5H
3  11/03/2020       15   17                        0            5S
4  11/03/2020       15   17                        0            6H

   Vehicle Count Payment Type (Cash or E-ZPass)
0              1                         E-ZPass
1             19                         E-ZPass
2             13                         E-ZPass
3              1                         E-ZPass
4              1                         E-ZPass
```

Use additional cells as needed to explore this data. Answer the following questions:

- How is the data collected? Is it automatic, or is there human involvement?
- What variable should be the *target variable* for this machine learning problem?
- What variable(s) could potentially be used as *features* to train the model?

- What are our assumptions about the features and the target variable, and the relationships between these? (For example: what times of day should be busy? What times of day will be less busy? What stretches of the Thruway might be especially congested - look at Google Maps?)
- Are there variables that should *not* be used as features to train the model, because of potential for data leakage?
- Are there any serious data problems that we need to correct before using the data for this purpose? Explain.

Satirical headline classification

Scenario

You are hired by a major social media platform to develop a machine learning model that will be used to clearly mark *satirical news articles* when they are shared on social media.

You consider using this dataset of 9,000 headlines from The Onion and 15,000 headlines from Not The Onion on Reddit. Link to OnionOrNot data

(This notebook shows how the data was compiled and processed.)

Read in data

This time, we'll retrieve the data from Github.

```
!git clone https://github.com/lukefeilberg/onion.git
```

```
Cloning into 'onion'...
remote: Enumerating objects: 10, done.ote: Counting objects: 100% (10/10), done.ote:
    Compressing objects: 100% (9/9), done.ote: Total 10 (delta 2), reused 0 (delta 0),
    pack-reused 0
```

```
df_headline = pd.read_csv("onion/OnionOrNot.csv")
df_headline.head()
```

```
                                                text  label
0  Entire Facebook Staff Laughs As Man Tightens P...      1
1  Muslim Woman Denied Soda Can for Fear She Coul...      0
2  Bold Move: Hulu Has Announced That 'Theyre Gon...      1
3  Despondent Jeff Bezos Realizes 'Hell Have To W...      1
4  For men looking for great single women, online...      1
```

Use additional cells as needed to explore this data. Answer the following questions:

- How is the data collected? Is it automatic, or is there human involvement?
- What variable should be the *target variable* for this machine learning problem?
- What variable(s) could potentially be used as *features* to train the model?
- What are our assumptions about the data?
- Are there variables that should *not* be used as features to train the model, because of potential for data leakage?
- Are there any serious data problems that we need to correct before using the data for this purpose? Explain.