I. Basic information for winner announcement

Please provide your preferred information for use in announcing the winners of the competition.

- Name (first and last name or first name and last initial): Jim King
- Hometown: Mill Valley, CA
- A recent picture of yourself or digital avatar (feel free to attach separately):
- Social handle or URL (optional): https://jimking100.github.io

II. Model documentation and write-up

Information included in this section may be shared publicly with challenge results. You can respond to these questions in an e-mail or as an attached file. Please number your responses.

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.

   I'm a successful real estate agent who happens to like data science. My Computer Science degree, MBA in Finance and 20 years in the tech field also help.

2. What motivated you to compete in this challenge?

   I love learning, a challenge and competition! The fact that the work may have a positive social impact is an added bonus.

3. High level summary of your approach: what did you do and why?

   I used a custom Named Entity Recognition (NER) model to identify cause and precipitating event keywords in the narrative section of the primary data. The custom keywords were organized into four major categories: method of fall (fall, trip, slip), care facility involvement, medical event involvement, and activity involvement.

   There are several reasons for selecting this approach:

   - Most of narrative is detailed in the other features, so the few words remaining often refer to the cause or precipitating event. The key is to identify "cause" keywords and a NER model is an excellent method.

   - A custom NER model allows for the very specific medical terms and events used in the narrative.

   - A NER model can be used to highlight cause factors in the text and provide real-time feedback to those entering and using narrative data.

4. Do you have any useful charts, graphs, or visualizations from the process?

   see attachments

5. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

```
#List of Entities and Patterns
patterns = [
        {"label": "TRIP", "pattern": "TRIP"},
        {"label": "TRIP", "pattern": "TRIPPED"},
        {"label": "TRIP", "pattern": "TRIPPING"},
        {"label": "TRIP", "pattern": "T'D"},
        {"label": "TRIP", "pattern": "T'D&F"},
        {"label": "TRIP", "pattern": "T/P"},
        {"label": "TRIP", "pattern": "STUMBLE"},
        {"label": "TRIP", "pattern": "STUMBLED"},
        {"label": "TRIP", "pattern": "TANGLE"},
        {"label": "TRIP", "pattern": "TANGLED"},
        {"label": "TRIP", "pattern": "STUB"},
        {"label": "TRIP", "pattern": "STUBBED"},

        {"label": "SLIP", "pattern": "SLIP"},
        {"label": "SLIP", "pattern": "SLIPPED"},
        {"label": "SLIP", "pattern": "SLIPPING"},
        {"label": "SLIP", "pattern": "S'D"},
        {"label": "SLIP", "pattern": "S'D&F"},
        {"label": "SLIP", "pattern": "S/P"},
        {"label": "SLIP", "pattern": "SLID"},
        {"label": "SLIP", "pattern": "WET"},

        {"label": "FACILITY", "pattern": "NURSING HOME"},
        {"label": "FACILITY", "pattern": "NH"},
        {"label": "FACILITY", "pattern": "REHAB"},
        {"label": "FACILITY", "pattern": "ECF"},
        {"label": "FACILITY", "pattern": "CARE"},
        {"label": "FACILITY", "pattern": "FACILITY"},
        {"label": "FACILITY", "pattern": "GROUP HOME"},

        {"label": "EVENT", "pattern": "SYNCOPAL"},
        {"label": "EVENT", "pattern": "SYNCOPE"},
        {"label": "EVENT", "pattern": "DIZZY"},
        {"label": "EVENT", "pattern": "PASSED OUT"},
        {"label": "EVENT", "pattern": "ALZHEIMER"},
        {"label": "EVENT", "pattern": "ALZHEIMERS"},
        {"label": "EVENT", "pattern": "PARKINSON"},
        {"label": "EVENT", "pattern": "PARKINSONS"},
        {"label": "EVENT", "pattern": "DEMENTIA"},
        {"label": "EVENT", "pattern": "DEMENTED"},
        {"label": "EVENT", "pattern": "LIGHTHEADED"},
        {"label": "EVENT", "pattern": "UNSTEADY"},
        {"label": "EVENT", "pattern": "SEIZURE"},
        {"label": "EVENT", "pattern": "PARKINSON"},
        {"label": "EVENT", "pattern": "BALANCE"},

        {"label": "ACTIVITY", "pattern": "WALKING"},
        {"label": "ACTIVITY", "pattern": "PLAYING"},
        {"label": "ACTIVITY", "pattern": "BENDING"},
        {"label": "ACTIVITY", "pattern": "WORKING"},
        {"label": "ACTIVITY", "pattern": "HIKING"},
        {"label": "ACTIVITY", "pattern": "CARRYING"},
        {"label": "ACTIVITY", "pattern": "MOWING"},
        {"label": "ACTIVITY", "pattern": "DANCING"},
        {"label": "ACTIVITY", "pattern": "ROLLERSKATING"},
        {"label": "ACTIVITY", "pattern": "SWIMMING"},
        {"label": "ACTIVITY", "pattern": "TRANSFERRING"},
        {"label": "ACTIVITY", "pattern": "LIFTING"},
        {"label": "ACTIVITY", "pattern": "FIXING"},
        {"label": "ACTIVITY", "pattern": "REACHING"},
        {"label": "ACTIVITY", "pattern": "STANDING ON"},
        {"label": "ACTIVITY", "pattern": "PICKING UP"},
        {"label": "ACTIVITY", "pattern": "PICK UP"},
        {"label": "ACTIVITY", "pattern": "GETTING UP"},
        {"label": "ACTIVITY", "pattern": "GET UP"},
        {"label": "ACTIVITY", "pattern": "GOT UP"},
        {"label": "ACTIVITY", "pattern": "GOING UP"},
        {"label": "ACTIVITY", "pattern": "GOING DOWN"},
        {"label": "ACTIVITY", "pattern": "GETTING OUT"},
        {"label": "ACTIVITY", "pattern": "GET OUT"},
        {"label": "ACTIVITY", "pattern": "GOT OUT"}

    ]
```

The code on the previous page associates the words or patterns with the custom defined entities (TRIP, SLIP, FACILITY, EVENT, ACTIVITY). The words or patterns were selected by manually reviewing the narratives to determine the key words and patterns to associate with which entities. Exploration of the top keywords was done using Yake (see example in the Community Code).

```
In [5]: %%capture --no-display
        for ind in df.index:
            text = df['narrative'][ind]
            doc = nlp(text)
            displacy.render(doc, style="ent")
            if ind > 500:
                break
```

94YOM FELL TO THE FLOOR AT THE NURSING HOME FACILITY ONTO BACK OF HEAD SUSTAINED A SUBDURAL HEMATOMA

86YOM FELL IN THE SHOWER AT HOME AND SUSTAINED A CLOSED HEAD INJURY

87YOF WAS GETTING UP ACTIVITY FROM THE COUCH AND FELL TO THE FLOOR SUSTAINED ABRASIONS TO ELBOWS ADMITTED FOR HEMORRHAGIC STROKE

67YOF WAS AT A FRIENDS HOUSE AND SLIPPED SLIP ON WATER THAT WAS ON THE FLOOR AND SUSTAINED A RIGHT RADIUS FX

One of the strong features of a NER model is the ability to apply it directly to the narrative text (see above code) to highlight the named entities and their associated label. This could be used to highlight the cause factors in the text providing more real-time feedback to those entering and using the data.

```
l]: def highlight_cells(val):
        color = ''
        if type(val) == str:
            if 'Activity - Walking/Hiking' in val:
                color = 'lightgray'
            if 'Products - Baths/Showers' in val:
                color = 'lightskyblue'
            if 'Products - Rugs/Carpets' in val:
                color = 'gold'
            if 'Facility - Nursing Home' in val:
                color = 'lawngreen'
            if 'Medical Event - Fainting' in val:
                color = 'lightcoral'
        return 'background-color: {}'.format(color)

    all_causes = all_causes.rename(columns={'cause': 'ALL CAUSES', 'pct': 'A PCT'})
    all_causes['SLIP CAUSES'] = slip_causes['cause']
    all_causes['S PCT'] = slip_causes['pct']
    all_causes['TRIP CAUSES'] = trip_causes['cause']
    all_causes['T PCT'] = trip_causes['pct']
    all_causes['FALL CAUSES'] = fall_causes['cause']
    all_causes['F PCT'] = fall_causes['pct']
    all_causes['SERIOUS CAUSES'] = serious_causes['cause']
    all_causes['R PCT'] = serious_causes['pct']

    all_causes.head(10).style.applymap(highlight_cells)
```

Since my award is for best visualization, the code above creates the colored dataframes used to create the attached visualizations.

6. Please provide the machine specs and time you used to run your model.
   - CPU (model): Apple M2
   - GPU (model or N/A): N/A
   - Memory (GB): 24GB
   - OS: Mac
   - Train duration: 2 minutes
   - Inference duration: less than a minute

7. Anything we should watch out for or be aware of in using your notebook (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

The Following Steps Need to Occur in the Terminal (1st Time Only) ¶

```
7]: # Create config file and train model in terminal using the stored training and validation data.
    # These steps are difficult to do in a notebook as specific configurations are machine and environment specific.

    # See https://spacy.io/usage/training

    # python -m spacy init fill-config base_config.cfg config.cfg
    # python -m spacy train config.cfg --output ./output --paths.train ./train.spacy --paths.dev ./val.spacy
```

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

   Yes, I used Yake for exploratory data analysis, an example is posted in the Community Code.

9. How did you evaluate the quality of your analysis and insights, if at all?

   First, the 'common sense' test – did the results seem to make sense. Second, running the NER model on the first 1000 narratives and then reviewing them for accuracy. The NER model's ability to highlight the keywords in the narratives makes this a relatively easy process (maybe an hour).

10. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

    I can't recall the exact things I tried, but I do recall the other methods I tried kept tripping up on the medical terms (e.g. a broken arm is a result of the fall and fainting is a cause of the fall – both are medical terms).

11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?

    In order to build on the work, a more exhaustive look at the 'cause' words in the narrative would increase the accuracy of the model. This is a manual process but is helped by using tools such as Yake to highlight the most frequently used keywords.

12. What simplifications could be made to run your solution faster without sacrificing performance?

    This already is a simple approach that performs very quickly.