EECS 598 Term Project

The Detroit Tigers have given us three interesting real-world data science projects. In this document, we'll discuss how to get started and describe scope and expectations for your term project.

Data loading

If you have signed and submitted the NDA, you will have access to SQL databases with their data.[1] We will be working with a PostgreSQL database hosted on a U of M server.[2] PostgreSQL is a mature, open-source relational database management system. Let's connect!

We will use the **LibPQ** and **DataStreams** packages to connect to the server and steam the data. You have been provided a codex on bookalive named "getting-started". Refer to the section "**Connecting to databases**" which contains functions that connect to each of the three databases. You can paste the functions in your notebook or make a module in your working folder that can be imported to use these functions. Once you have loaded the data using stream(), you are good to go!

Other tools

We discussed the **Flux** package in class. It is an elegant Julia machine learning package. You can use it to efficiently implement various neural networks. You can also use the models/codexes derived in the class - for example, the nearest subspace classifier, matrix completion algorithms, etc. OR you can also write the code from scratch on your own!

Installing Jupyter. To run the notebooks that we provide you and the ones that you will make, you will need to install Jupyter on your laptops (or any system you choose to use).

Add the package "**IJulia**" using Pkg.add("IJulia"). Then you can just open a new Jupyter notebook as shown here:

```
using IJulia
notebook()
```

---

[1] Why SQL and not CSV? While CSV files are pleasantly simple, they have downsides for large datasets. First, for everyone to have their own local copy of the CSV files, we must duplicate hundreds of megabytes of data hundreds of times. Ultimately each team will only use a subset of this data. All that duplication is wasteful and does not scale well! Good data is also valuable, may be updated frequently, and is often sensitive. With CSVs, if a copy of the file makes it to the wrong person or the public internet (yikes!), it may remain in the wrong hands forever. These days, large datasets are often stored as SQL-compliant databases and hosted from servers. This circumvents needless duplication - there is no need for each user of the data to have their own complete copy. A SQL database can also be configured to require authentication, and the database administrator can revoke access once it is no longer required for a particular user.

[2] Fun fact: arguably the first implementation of this relational model was built at U of M, in 1969!

Milestones and Deliverables

There will be two major milestones, each with its own deadline.

Milestone 1

Required. Due November 21, 2018.

The aim of the first milestone is to play with the data in SQL and understand your chosen task. You should develop an understanding of which **models** work best for the chosen goal/task, what **loss functions** are most appropriate, and what **metrics** would be best for evaluating outputs fairly. This milestone has two deliverables.

M1 Deliverable 1: Report

Submit a 1-2 page report (not including figures) summarizing:

- Algorithm(s) used
    - If the team has n members, you should demonstrate n+1 working models for your task
- Math of the loss functions you are trying to minimize
    - For example, if you are using a neural network and MSE loss, write down the math equations that take the input to the output of the network and how the loss is computed related to this mapping..
- Evaluation score/metric
    - The choice of metric is a key aspect of evaluating your algorithm. Depending on the task, you may choose MSE, cross entropy, accuracy, precision, recall, etc.
    - You should devise one metric on your own - it can be as simple as using a combination of metrics mentioned above. We leave this part open to you.
    - Comparison of the metrics used: what is a better metric and why?
- Timing statistics: how long does it take to train and test

Be as clear as possible. We want to see your data analysis skills through this project as you apply course concepts to real-world problems!

In addition to the above-mentioned points, you could also try to address the challenges you faced and how you solved them. It is not mandatory to answer these questions, but addressing them would definitely strengthen your report :)

- Was any encoding or transformation of features necessary? If so, what encoding/transformation did you use?
- Which features had the greatest impact on the chosen task? How did you identify these to be most significant? Which features had the least impact, and how did you know?

- If you trained a model, how did you train it? During training, what issues concerned you?

M1 Deliverable 2: Code

Refer to the file Creating-a-module-in-Julia.pdf in the main folder on Canvas. We strongly encourage you to create a module to hold all your functions. You can then load the module in your notebook and call your functions accordingly. Your code should be well formatted: **code quality will be judged**. **Also, all results must be reproducible.**

In addition to the code module, submit a Julia notebook, which when run from start to end will perform all the steps required - starting from data loading, pre-processing, and getting the required numbers and plots. The notebook should have three code cells:

1. **Data cell**. When this cell is executed, the data you will be using are loaded and pre-processed.
2. **Training and analysis cell.** The code in this cell performs your analysis, including any model training. If you are training a model, you **must** save the model parameters or weights so that we can load the weights and directly run the tests. (You may include a second analysis cell that simply loads the parameters and performs classification/completion.)
3. **Results cell.** This cell processes the output and generates any plots you would like to use to explain your results.

While there should be just three code cells, we encourage you to add markdown cells between them to explain each of the three steps in detail (especially your results).

Milestone 2

Optional. Due November 29, 2018.

In the first milestone, it was ok to get mediocre-to-reasonable performance based on whatever metric you used. The next step is to make your model the best in class! Using techniques from class and your own research, tweak your analysis to improve its performance. This may involve more sophisticated encoding or handling of outliers in the data, adding some regularization technique, tuning model parameters, and other improvements.

M2 Deliverable 1: final report

This report should be an expanded version of the M1 Deliverable 1 report. Aim for a 1-2 page report (again, not including figures). Discuss the modifications you made, why they work, and how well they work. Also address how much further you think your method could go with “perfect” tuning, and what other methods might beat yours.

M2 Deliverable 2: final code

Your report should be accompanied by a final version of your code. As with M1 Deliverable 2, we encourage you to submit a Julia module containing all your functions and types (if necessary). Also turn in a final version of the three-code-cell notebook that loads data, performs analysis, and produces results. **All results must be reproducible.**

If you develop an algorithm that attains an evaluation score within 10% of the top score in the class and also runs within 10% of the fastest time -- you will automatically get a 50% bonus. The leaderboard will be made public after all the submissions have been evaluated so that you know what the top score is. If you did not make the cut in this regard, you may then choose to resubmit an improved algorithm that is within 10% of the top score in terms of the metric that we will specify and the run time criterion and if you succeed, you too will get the 50% bonus.

Leaderboard

For each task, we will announce standard metrics to be used for fair comparison between submissions. Stay tuned for details.