



**Team Project Brief: DrivenData**

**MMAI/MMA/GMMA 869 (Machine Learning and AI)**

**Dr. Stephen Thomas**

**Version 1 (June 17, 2021)**

# Executive Summary

“Help the world.” - Uncle Steve (and probably some other people)

You are to enter a live competition on <https://www.drivendata.co> and use your skills to help make the world a better place.

# The Competitions

DrivenData hosts data science competitions to build a better world. In particular, they host competitions related social impact, in the areas of health, conservation, development, equality, disaster recovery, education, and more.

There are three live competitions that we will consider:

* [Pump it Up: Data Mining the Water Table](https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/): predict which water pumps in Tanzania are faulty.
* [Flu Shot Learning: Predict H1N1 and Seasonal Flu Vaccines](https://www.drivendata.org/competitions/66/flu-shot-learning/): predict who will get vaccinated
* [Richter's Predictor: Modeling Earthquake Damage](https://www.drivendata.org/competitions/57/nepal-earthquake/page/134/): predict the level of damage to buildings caused by the [2015 Gorkha earthquake in Napal](https://www.youtube.com/watch?v=WwIw1-voHKQ&t=1s).

Each team has been randomly assigned a competition as follows:

|  |  |  |
| --- | --- | --- |
| Cohort | Team | Competition |
| MMA 2022W Section 1 | Bremner | Water Pumps |
| MMA 2022W Section 1 | Carlton | Vaccines |
| MMA 2022W Section 1 | Dundas | Earthquake |
| MMA 2022W Section 1 | Eglington | Water Pumps |
| MMA 2022W Section 1 | Finch | Vaccines |
| MMA 2022W Section 1 | Gerrard | Earthquake |
| MMA 2022W Section 1 | LAwrence | Water Pumps |
| MMA 2022W Section 1 | Richmond | Vaccines |
| MMA 2022W Section 1 | Wellington | Earthquake |
| MMA 2022W Section 2 | Avenue | Water Pumps |
| MMA 2022W Section 2 | Islington | Vaccines |
| MMA 2022W Section 2 | Jarvis | Earthquake |
| MMA 2022W Section 2 | Kipling | Water Pumps |
| MMA 2022W Section 2 | Ossington | Vaccines |
| MMA 2022W Section 2 | Parliament | Earthquake |
| MMA 2022W Section 2 | Spadina | Water Pumps |
| MMA 2022W Section 2 | Yonge | Vaccines |

# Your Project

Your mission consists of two parts: exploratory data analysis and machine learning.

## Part 1: Understand the Patterns and Trends

Teams will use exploratory data analysis and unsupervised ML techniques to quantify and analyze the trends in the dataset. For example, you can:

* Use plots and graphs to tell a story and discover correlations/trends/outliers
* Use association rule learning to uncover patterns and trends
* Use cluster analysis to build clusters of {water pumps, people, buildings}, and describe the clusters and build personas.

There are no strict requirements here, other than to use the tools available to better understand the data. Doing so will be interesting in its own right, but will hopefully also help guide you in Part 2.

## Part 2: Compete!

Teams will use the data provided by the competition to build and assess a model, as described in the competition. Teams will submit their predictions to the competition, iterate, refine, and improve.

Teams shall:

* Make at least one submission to the competition (but more likely, dozens)
* Try many different models/preprocessing techniques/hyperparameter tuning options
* For your “best” model:
  + Describe/quantify the performance of the model using confusion matrices and the macro F1 score.
  + Describe the drivers (i.e., feature importances) of your model’s performance. What did your model “learn?”
* Specify what you would need (in terms of data, compute power, algorithms, etc.) to improve the model’s performance if you had more time and money.

# Deliverables

Teams will be responsible for the following deliverables:

* A 10-minute live presentation that includes:
  + The insights/results for Part 1: a thorough exploratory data analysis (EDA) of the data including summary statistics, charts/plots, etc.
  + The modeling results and analysis for Part 2
  + Next steps/Future Work
  + Conclusion/Summary
* An accompanying report is not necessary.
* You may submit your source code, but it is not necessary and it will not be marked.

## Presentation Tips

* This is a short presentation. Don’t linger on unimportant stuff. Focus on the juicy bits.
  + Don’t include an agenda slide. This presentation is not long enough to need one; and spending time on an agenda is not worth the time.
  + Don’t include team introductions. (“*Hi, I’m Steve, and this Bill, and over there is Mary, and there’s Hector, and then we have Mona, and finally my dog Roofus. We’re part of Team Toronto and we have been working on this project together.”)* It takes too long and is not worth the time. (In the past, teams have spent 1 minute introducing themselves. That’s almost 10% of the entire presentation spent on fluff!)
  + Don’t spend any time on the title slide – just get started. (In the past, teams have spend 1-3 minutes with the title slide showing, talking about “meta” topics, like “*you know, we really had a great time in this project, and I’m happy to be here, and in fact my father used to work at a pharmacy, but then he moved into retail, but I still love the movies, you know, and my teammates, uh, my teammates and I are excited to share our results, and I wanted to thank Uncle Steve for letting us use his code, and I’m kinda nervous right now which is why I’m talking a lot hahaha. Can you see my screen?*”) The clock is ticking and everyone has limited patience. They want you to get started - so just get started.
* Spend less time on *how* you did something, and more time on *what you learned*.
  + E.g., less “*we loaded the data in pandas, dropped rows with missing values, installed the XYZ package, and ran it for 2 hours, and calculated the metrics*” and more “*we found that Kingston residents are 34% more likely to switch*.”
  + E.g., less “*we did an association rules algorithm and found 77 rules, and the top rule is a rule with 66% support*” and more “*We found that eggs are bought with milk 66% of the time.*”
  + Fewer uses of the phrases “*clustering algorithm*” or “*association rules*” or “*training*.” Just focus on the insights.
  + Put the “how” in the Appendix:
    - Cleaning steps
    - Package details
    - Modeling details (algorithm choice, hyperparameter values, etc.)
    - Etc.
* Make your presentation an interesting story. Not just “*we did this, then this, then that, then then then*.”
* Be creative and have fun!
  + Pictures are better than words
  + Graphs are better than words
  + Charts are better than words
  + Tables are better than words

The target audience for this presentation is your average MMA student: a tech-saavy manager who wants trustworthy, detailed insights. The audience is not afraid of a little tech talk, but the focus should be on the results, insights, and conclusions, not the method.

* End strong. Don’t finish with a slide that just says “Questions??” That’s a wasted opportunity. Give us a useful TLDR.
* You will not be graded on your performance relative to other competitors.

### Rubric

Up to four marks will be available in the following categories.

* Part 1
  + EDA Quality. Told a story, uncovered insights, captivated the audience
  + EDA Style. Effective fonts, colors, graphs, visuals
  + EDA Focus. Focused on insights and results, not tech process and methodology
* Part 2
  + Overall effort. High quantity/breadth of ML techniques attempted.
  + Description of best model. Performance, feature importances, analysis of errors
  + Analysis of model prediction errors. In depth look at the mistakes your model makes. Why – are there any patterns? Does it reveal anything about the data, or the algorithms?
* Next steps. Description of what you would try if you had more time/budget.
* Lessons learned. Inclusion of concise and helpful lessons learned during the project.
* Clarity of presentation. Overall clarity/understandability of presentation.
* You may earn bonus points as follows:
  + Highest competition score in section: 2 bonus points

# Language and Platform

Teams may use any programming language and IDE/platform/tool they wish.

I recommend using the Python programming language (using standard packages like pandas and scikit-learn) on the Jupyter Notebook platform. Google Colab will be perfect for this project.

For tips on learning Python and Jupyter, please see the “Programming Languages and Tools” section of the course portal.

# FAQ

Can we use your example Python Notebooks in your GitHub repository?

Absolutely! Yes. Please use them as a launching off point.

Is there a Subject Matter Expert (SME) to whom we can ask questions about the data?

No, but Uncle Steve and Meghan are here to help.

How did students last year do on this project?

This is the first year that we’ve ever done this project. It’s all an experiment! Nobody knows what patterns we will find, or whether a good model is even possible.

My code has an error. What should I do?

First, you should understand the error. Read the whole thing. What is it telling you? The error message will often lead you directly to the answer if you just read it carefully.

If the error message isn’t clear, or you don’t know how to solve it, you should Google the error. Google is by far your best friend. All the pros use Google. Google knows what to do. You probably aren’t the first person to have this error.

If you are can’t figure it out by Googling, you should consult your teammates. Teams that learn together, stay together!

If you are stilling having the error, you should read your code carefully. You know what they say: 3 hours of debugging can prevent 3 minutes of reading your code! (Or something like that. It’s a joke.)

Next, you should ask the TA via email. When you ask the TA, please include the following:

* What exactly is the error message?
* What have you tried so far to fix your code?
* What kind of data is in the data frames/variables that are involved (if any)?
* What have you Googled? What documentation have you read?
* What will you try if you can’t get this to work? (What is Plan B?)

The more information you give the TA, the higher the probability that the TA can help you.

Finally, if all else fails, email Uncle Steve and/or bring your issue to the next office hours with Uncle Steve.

*How can improve the performance of my model?*

* Feature engineering is your friend
  + In particular, there are a lot of categorical features with many unique values. What can we do? (Hint: check out the package [category\_encoders](https://contrib.scikit-learn.org/category_encoders/).)
* The data is probably imbalanced, so you can do things like:
  + add a weights
  + down sample
  + up sample
* Boosting is your friend
  + Adaboost, XGBoost, LightGBM, catboost
* Hyper-parameter tuning is your best friend
  + But don’t overfit! Cross validation is your friend.