

**Team Project Brief**

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

**Dr. Stephen Thomas**

**Version 1 (October 17, 2022)**

# Executive Summary

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

Teams are to enter a live machine learning competition on [DrivenData](https://www.drivendata.org/competitions/) or [Zindi](https://zindi.africa/learn/fraud-detection-in-electricity-and-gas-consumption-challenge-tutorial) and use their skills to help make the world a better place.

The main objectives of this project are for teams to:

1. Practice pushing ML models “to the limit” by exploring different techniques in preprocessing and cleaning, feature engineering, feature selection, data augmentation, ML algorithms, hyperparameter tuning, and ensembles.
2. Learn the effect of building a model on one dataset and testing it on another.
3. Practice splitting a complicated project amongst themselves in a tight timeline.

# The Competitions

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

There are several live competitions that we will consider in this project.

* [DrivenData: 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.
  + This competition is challenging because of dirty/noisy/missing data and contains a multi-class objective.
  + Performance metric: Accuracy
* [Driven Data: Flu Shot Learning: Predict H1N1 and Seasonal Flu Vaccines](https://www.drivendata.org/competitions/66/flu-shot-learning/): predict which Americans will get vaccinated.
  + This competition is challenging because it is a multi-label/multi-target objective and has lots of missing data.
  + Performance metric: AUC
* [Driven Data: 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 Nepal](https://www.youtube.com/watch?v=WwIw1-voHKQ&t=1s).
  + This competition is challenging because the data is large, it has a multi-class objective, and the most valuable features are categorical with many categories/levels.
  + Performance metric: Micro F1 Score
* [Zindi: Fraud Detection in Electricity and Gas Consumption](https://zindi.africa/learn/fraud-detection-in-electricity-and-gas-consumption-challenge-tutorial): predict which clients are involved in fraudulent activities.
  + This competition is challenging because the data is large, and the features need to be aggregated at the client level.
  + Performance metric: AUC
* [Zindi: Nigeria Loan Default Prediction](o%09https:/zindi.africa/competitions/data-science-nigeria-challenge-1-loan-default-prediction/data): predict which bank loans will default.
  + This competition is challenging due to class imbalance and because the features need to be aggregated at the customer level.
  + Performance metric: Error (i.e., 1 – Accuracy)

I have assigned each team to one of the competitions as follows:

|  |  |  |
| --- | --- | --- |
| Cohort | Team | Competition |
| MMA 2023S | Chernoff | Pump |
| MMA 2023S | Douglas | Flu |
| MMA 2023S | Dunning | Earthquake |
| MMA 2023S | Ellis | Fraud |
| MMA 2023S | Goodes | Loans |
| MMA 2023S | Stirling | Loans |
| MMA 2023S | Aberdeen | Pump |
| MMA 2023S | Albert | Flu |
| MMA 2023S | Alfred | Earthquake |
| MMA 2023S | Barrie | Fraud |
| MMA 2023S | Frontenac | Loans |
| MMA 2023S | Union | Loans |
| MMAI 2023 | Broadview | Pump |
| MMAI 2023 | Chester | Flu |
| MMAI 2023 | College | Earthquake |
| MMAI 2023 | Dufferin | Fraud |
| MMAI 2023 | Greenwood | Loans |
| MMAI 2023 | Landsdowne | Fraud |
| MMAI 2023 | Wellesley | Loans |

# The Project

The primary object of the project is to throw caution to the wind and compete to win. Teams are to go “all-out” in trying to climb the competition leaderboards.

## Part 1: The Competition

Teams will use the data provided by the competition to build and assess a machine learning 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 techniques for preprocessing, cleaning, feature engineering, ML algorithms, hyperparameter tuning, and all the other topics we learn in this course.
* Compete! Do whatever you can to move up the rankings. Try things that should work, and try things that shouldn’t work. Experiment, iterate, and go for it.
* Feel free to do as much data exploration/EDA as necessary to understand 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.

I recognize that ML competitions are not always realistic examples of ML in business settings (what’s the point in spending an extra $50,000 for a model that is only 0.05% better?). However, I do believe there is tremendous learning value when you try to push something to its absolute limits. The competitive nature of these competitions will motivate you to work hard and try many new techniques quickly, thereby allowing you to learn what works and what doesn’t on your own.

# Deliverables and Rubric

Teams will be responsible for a 12-minute live presentation that outlines the story of their competition journey. Please include:

* Screenshot of your best ranking on the competition website.
  + Note: you will not be graded on the actual performance/ranking in the competition (although I wish you the very best of luck)
* 10%: What cleaning and preprocessing steps did you try? Which worked, which didn’t?
* 10%: What feature engineering and selection steps did you try? Which worked, which didn’t?
* 10%: Which ML algorithms did you try? How well did they work?
* 10% What hyperparameter tuning procedure did you try? What range of values did you consider? How much did they help performance?
* 10%: What else did you try? Any particular insights or “Eureka!” moments?
* 10%: Describe your best model/submission in more detail, such as:
  + Describe/quantify the model’s performance using confusion matrices and the associated metrics.
  + If possible, describe the drivers (i.e., feature importances) of your model’s performance. What did your model “learn?”
* 10%: Next steps. Description of what you would try if you had more time/budget. 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. Be as specific as possible.
* 10% Lessons learned. Inclusion of concise and helpful lessons learned during the project.
* 20% Clarity of presentation. Overall clarity/understandability of the presentation, including slide design and oral delivery. Ability to answer questions during the Q&A portion of the presentation

An accompanying report is not necessary.

## 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 spend time on team member introductions. (“*Hi everyone, 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 1-2 minutes 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 spent 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.
* If necessary, create an Appendix with additional information:
  + Cleaning steps
  + Package details
  + Modelling details (algorithm choice, hyperparameter values, etc.)
  + Etc.
* 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-savvy manager who wants to learn what you learned.

# Language and Platform

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

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 Cecilia are here to help.

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 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. You probably aren’t the first person to have this error.

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

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

*How can I 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/) or use an algorithm with built-in categorical support like Histogram-based Gradient Boosting, LGBM, or CatBoost.)
* Boosting is your friend
  + XGBoost, LightGBM, CatBoost
* The data is probably imbalanced, so you can do things like:
  + add weights
  + downsample
  + upsample
* Hyperparameter tuning is your best friend
  + GridSearchCV, Optuna
  + But don’t overfit! Cross-validation is your friend.