

**Team Project Brief**

MMAI 869 (Machine Learning and AI)

Dr. Stephen Thomas

*Updated: May 6, 2025*

# Overview

Teams will compete in the annual *MMAI 869 Machine Learning Competition* to obtain glory, bragging rights, hard-earned knowledge, and course credit. Each year’s competition is different. The theme of this year’s competition is space travel:

**The year is 2912. Can you predict which spaceship travelers will be transported to an alternate dimension?**

The main objectives of this project are for teams to:

1. Push ML Models to the Limit: Explore techniques in data cleaning, feature engineering, feature selection, data augmentation, ML algorithms, hyperparameter tuning, ensembles, and more.
2. Cross-Dataset Application: Understand the impact of building a model on one dataset and applying it to another.
3. Collaborative Project Management: Practice dividing a complex project among team members within a tight timeline.

# The Project

Welcome to the year 2912, where your data science skills are needed to solve a cosmic mystery[[1]](#endnote-1). We've received a transmission from four lightyears away and things aren't looking good.

The *Spaceship Titanic* was an interstellar passenger liner launched a month ago. With almost 13,000 passengers on board, the vessel set out on its maiden voyage transporting emigrants from our solar system to three newly habitable exoplanets orbiting nearby stars.

While rounding Alpha Centauri en route to its first destination—the torrid 55 Cancri E—the unwary Spaceship Titanic collided with a spacetime anomaly hidden within a dust cloud. Sadly, it met a similar fate as its namesake from 1000 years before. Though the ship stayed intact, almost half of the passengers were transported to an alternate dimension!

To help rescue crews and retrieve the lost passengers, you are challenged to predict which passengers were transported by the anomaly using records recovered from the spaceship’s damaged computer system. Help save them and change history!

## The Competition

Teams will be provided with a training dataset of 8,693 passengers. The features of the dataset include the passengers’ age, their home planet, which cabin they were staying in, their final destination, how much money they spent on the ship, and other details. The training dataset also indicates whether each passenger was transported to another dimension or not.

Teams will use the training dataset to build ML models that can predict whether a passenger was transported.

Teams will also be provided with a competition test dataset containing features of an additional 4,277 passengers. Teams will use their models to predict whether each of the passengers in the test set were transported. Teams will save the models’ predictions to a CSV file, and submit it to to the competition website:

[Kaggle: MMAI 869 2026](https://www.kaggle.com/t/b0bc0790cde44458aa2ddc5f46d982a8)

The competition website will automatically score their predictions (as it knows the correct answers) using the accuracy metric and will add the team’s score to the competition leaderboard.

Teams shall:

* Make at least one submissions. Teams will likely submit many more than one submission as they iterate, refine, and improve their ideas. Teams can submit up to 10 times per day.
* Try many different techniques for preprocessing, cleaning, feature engineering, ML algorithms, hyperparameter tuning, and all the other topics we learn in this course.
* Try at least one non-tree-based ML algorithm, such as KNN, Logistic Regression, Neural Networks, or SVM. (Teams can try as many tree-based algorithms as they wish, such as Decision Trees, Random Forests, LGBM, XGBoost, CatBoost, etc.)
* 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 three months to build a model that is only 0.05% better?). However, I believe there is tremendous learning value when you try to push something to its absolute limits. I hope that the competitive nature of this project will motivate teams to work hard and try many new techniques quickly, thereby allowing them to learn what works and what doesn’t.

# Deliverables and Rubric

Teams will be responsible for a 12-minute live presentation that outlines the story of their competition journey. An accompanying report is not necessary.

### Presentation Requirements

* **Final Leaderboard Accuracy Score**: Include the accuracy score of your best model.
  + Note: You will not be graded on the actual performance/ranking in the competition (although I wish you the very best of luck).
* **Distinguish Scores**: When discussing your various models, clearly differentiate between a model’s cross-validation score (CV score), which you estimated yourself, and its leaderboard score (LB score), which Kaggle calculates on the competition test data.

### Grading Criteria

1. **Data Cleaning and Preprocessing (10%)**:
   * Describe the cleaning and preprocessing steps you tried.
   * Explain which steps worked and which didn’t.
2. **Feature Engineering and Selection (10%)**:
   * Detail the feature engineering and selection steps you tried.
   * Discuss the effectiveness of each step.
3. **Machine Learning Algorithms (10%)**:
   * List the ML algorithms you tried, including at least one non-tree-based algorithm.
   * Evaluate the performance of each algorithm using cross-validation.
   * Compare each model’s CV score with its LB score.
4. **Hyperparameter Tuning (10%)**:
   * Describe the hyperparameter tuning procedures/algorithms you tried.
   * Specify the range of values considered and their impact on performance.
   * Present the final/best values.
5. **Innovative Approaches and Insights (10%)**:
   * Highlight any additional techniques or approaches you tried.
   * Share any particular insights or “Eureka!” moments.
6. **Detailed Model Description (10%)**:
   * Provide a detailed description of your best model/submission.
   * Quantify the model’s performance using a confusion matrices and associated metrics.
   * Describe the drivers (i.e., feature importances) of your model’s performance.
   * Show at least three training instances that your model predicted correctly and three that it predicted incorrectly. Draw insights from these examples.
7. **Next Steps (10%)**:
   * Describe 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.
8. **Lessons Learned (10%)**:
   * Include concise and helpful lessons learned during the project.
9. **Clarity of Presentation (20%)**:
   * Assess the overall clarity and understandability of the presentation, including slide design and oral delivery.
   * Evaluate the ability to answer questions during the Q&A portion of the presentation.

## 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. Everyone wants 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 MMAI 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. Kaggle Notebooks or 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.

See the course portal for the link to the competition website.

# 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. You can also try ChatGPT or ther GenAI tools. They are also very helpful!

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, RandomizedSearchCV, Optuna
  + But don’t overfit! Cross-validation is your friend.

1. Acknowledge of this competition’s outline and dataset goes to Google LLC. [↑](#endnote-ref-1)