

# Advisor Meeting Talking Points

Ryan MacKellar – HNR 499 Senior Project

---

## 1. Start with the Big Picture

*“AI is really just math in action. What looks like a ‘smart’ model is, underneath, linear algebra and calculus repeatedly applied to data. Libraries like TensorFlow or PyTorch are just tools that automate these calculations. My project shows how the math translates directly into code and working models.”*

---

## 2. Linear Algebra → Data Representation

- Every dataset can be represented with vectors and matrices.
  - In MNIST, each digit image is a  $28 \times 28$  grid of pixels → a matrix.
  - Neural networks use matrix multiplications at every layer: input  $\times$  weights = output.
  - Dimensionality reduction techniques like **PCA** come in when datasets are large. PCA finds the most important directions of variance and compresses the data while keeping its structure.
- 

## 3. Calculus → Learning from Mistakes

- Training a model means minimizing error (the “loss function”).
  - **Gradients** are the math that tells us how much each weight contributes to the error.
  - With **gradient descent**, the model shifts its weights slightly in the direction that reduces error the fastest.
  - Backpropagation is just the chain rule applied over and over, layer by layer, to calculate these gradients.
- 

## 4. Probability & Statistics → Making Predictions

- When the model predicts a digit, it doesn’t give a single answer — it gives probabilities (e.g., 90% chance it’s a “3”).
- Statistics helps us validate that the model isn’t just memorizing, but generalizing.

- Concepts like accuracy, precision, recall come from stats and let us judge whether the model is performing well.
- 

## 5. Coding → Automating the Math

- **NumPy** handles vectors and matrices.
  - **Scikit-learn** lets us apply PCA, regression, clustering with just a few lines of code.
  - **TensorFlow/PyTorch** automate gradient calculation and optimization.
  - **Matplotlib** lets us visualize learning curves or show PCA projections.
  - Example: in TensorFlow, loading MNIST is one line of code. But behind that line is matrix math and calculus happening thousands of times per second.
- 

## 6. Why MNIST Works Perfectly

*"I chose MNIST because it's the standard 'hello world' of machine learning. It's simple enough to work with but complex enough to demonstrate how the math drives the code. Each image is a perfect case study in turning raw data into vectors, applying matrix multiplications, optimizing with gradients, and validating results with statistics."*

---

*"So the narrative I'm building is: data starts as numbers → linear algebra organizes it → calculus adjusts weights through gradient descent → probability and statistics validate the results → and coding libraries tie it all together. My manual will walk through each step with math, code, and examples."*