Beyond Basics with **LingoRank**: Elevating French Text Analysis using Machine Learning

Data Science and Machine Learning

Detecting the difficulty level of French texts
Team: UNIL_Zurich

Matteo Frison   Takaaki Kishida

December 20, 2023

Our Project Goals

Enhancing Language Learning

- LingoRank is committed to revolutionizing language learning by accurately assessing the difficulty level of French texts for English speakers.

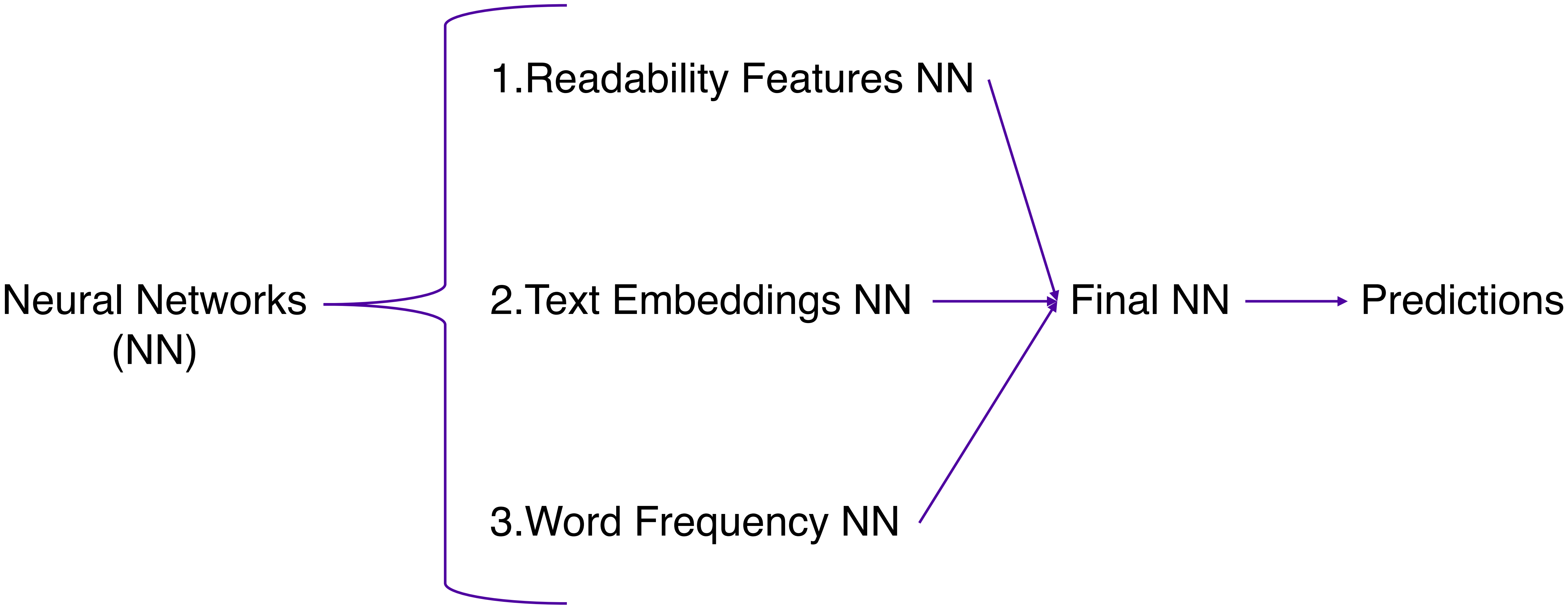
Personalized Learning

- Our primary goal is to enhance the learning experience by matching readers with texts that align with their current proficiency level.
- This approach ensures that learners are engaged with material that is challenging yet appropriate for their skill level.

Data Overview

- Source
  - The dataset consists of French texts of varying complexity
- Labels
  - Texts are labeled with language proficiency levels, ranging from A1 (beginner) to C2 (advanced).
- Volume
  - Training Data: 4,800 entries, 3 features each (id, sentence, difficulty).
  - Unlabeled Test Data: 1,200 entries, 2 features each (id, sentence).

Methodology Overview



Preprocessing Steps

1. Data Preparation

- **NN Readability feature**
  - (X) Avg sentence length, Avg word length, Avg syllable count per word, Nb of word with 1 to 3 syllables, Nb of words with 4 syllables, Nb of long (>10) words, Nb of {NOUN, AJD, ADV, ADP, PRON, DET, VERB} using spaCy "fr_dep_news_trf"
  - (y) Onehot Encoding
- **NN CamemBert**
  - (X) Tokenized sentences with CamembertTokenizer. Max length = 250.
  - (y) Label Encoding
- **NN Word Frequency**
  - (X) Tokenized sentences spaCy "fr_dep_news_trf" (with stopwords!). TfidfVectorizer with ngram_range=(1, 1).
  - (y) Onehot Encoding
- **Final NN**
  - (X) Predictions from the 3 NN above.
  - (y) Onehot Encoding

2. Data Splitting

- Initially, split into training and validation sets with a 70-30 ratio for the first 3 NN.
- Then, train the final model with a 80-20 ratio.
- Re-train the first 3 model with a 99.9-0.01 ratio.

Core Model Architecture and Training (1/4)

Model Architecture

- **Readability Feature**: Analyzes linguistic features such as sentence length and syllable count using pyphen and spaCy.
- Input Layer→BatchNormalization Layer→ Dense Layer (256 neurons, relu activation, L2)→Dropout Layer (0.1)→ Dense Layer (128 neurons, relu activation, L2)→Dropout Layer (0.1)→Dense Layer (6 neurons, softmax activation, L2)

Training Process

- **Optimization**: Applies AdamW optimizer and a linear learning rate scheduler. Lr = 3e-5
- **Training Details**: Conducts training over 100 epochs. Batch size = 16
- **Validation**: Evaluates model accuracy on a separate validation set.

Core Model Architecture and Training (2/4)

Model Architecture

- **CamemBERT Model**: Uses CamembertForSequenceClassification which is a CamemBERT Model transformer with a sequence classification/regression head on top. Model trained on 138GB of French text
- From **CamemBERT: a Tasty French Language Model:**
- “Similar to RoBERTa and BERT, CamemBERT is a multi-layer bidirectional Trans-former”
- “CamemBERT uses the original architectures of BERTBASE (12 layers, 768 hidden dimensions, 12 attention heads, 110M parameters)”

Training Process

- **Optimization**: Applies AdamW optimizer and a linear learning rate scheduler. Lr = 3e-5
- **Training Details**: Conducts training over 4 epochs. Batch size = 16
- **Validation**: Evaluates model accuracy on a separate validation set.

Core Model Architecture and Training (3/4)

Model Architecture

- **Word Frequency NN**: TF-IDF to use the importance of a word as a text difficulty measure.
- Input Layer→BatchNormalization →Layer Dropout Layer (0.1)→Dense Layer (6 neurons, softmax activation, L2)

Training Process

- **Optimization**: Applies AdamW optimizer and a linear learning rate scheduler. Lr = 3e-5
- **Training Details**: Conducts training over 60 epochs. Batch size = 16
- **Validation**: Evaluates model accuracy on a separate validation set.

Core Model Architecture and Training (4/4)

Model Architecture

- **Final Model**:
- Input Layer→BatchNormalization Layer→Dropout Layer (0.1)→Dense Layer (128 neurons, relu activation, L2)→Dropout Layer (0.1)→Dense Layer (6 neurons, softmax activation, L2)

Training Process

- **Optimization**: Applies AdamW optimizer and a linear learning rate scheduler. Lr = 3e-5
- **Training Details**: Conducts training over 100 epochs. Batch size = 16
- **Validation**: Evaluates model accuracy on a separate validation set.

Model Evaluation and Final Prediction

Evaluation and Prediction

- **Ensemble Approach**: Combines outputs from BERT, TF-IDF, and feature-based models for robust predictions.

- **Final Classifier Model**: Uses a concatenated output to predict difficulty levels with a custom TensorFlow model.

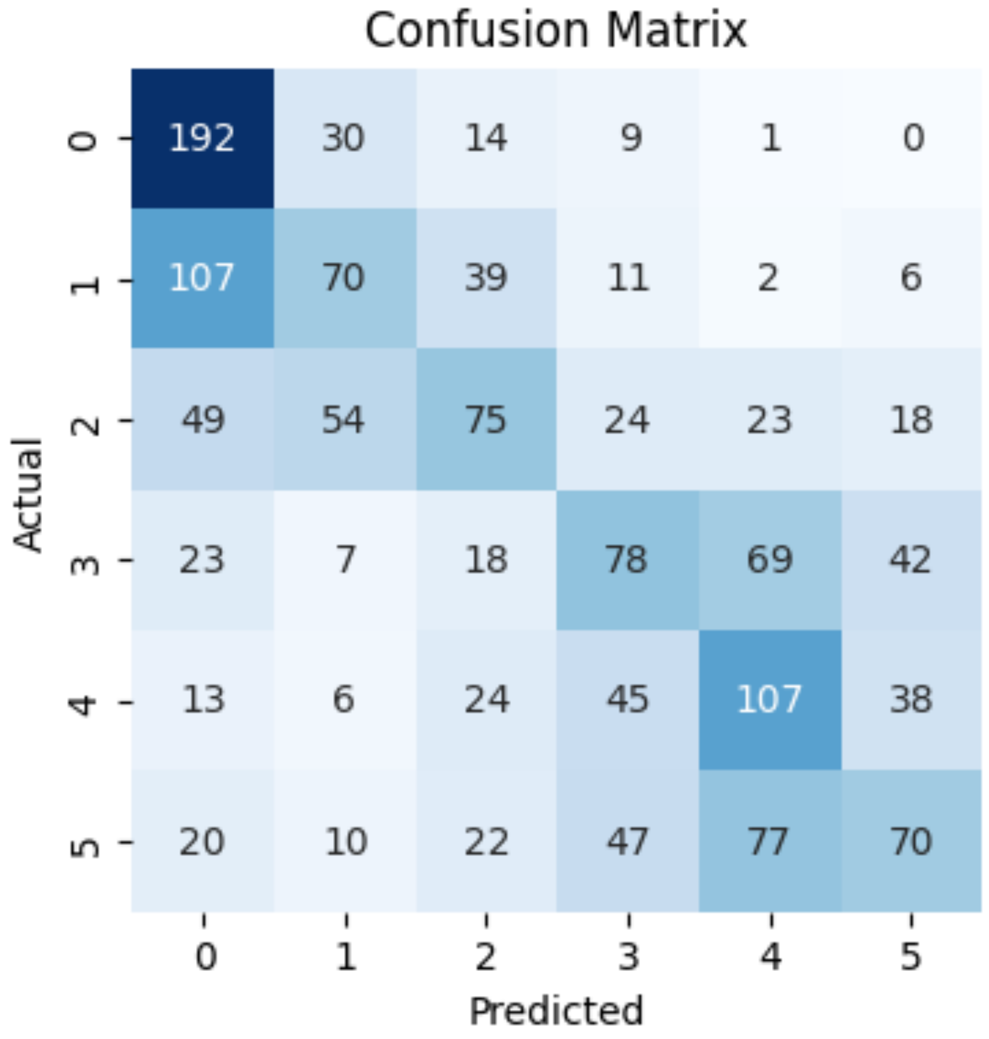
Output and Deployment

- **Performance Visualization**: Employs confusion matrices to showcase model accuracy.

- **Result Exporting**: Predicts difficulty levels for new sentences.

Results (1/4)

- **Readiblity features NN**
- Our model achieved an accuracy of 41.1%
- Strongest at identifying beginner, particular A1. Not too bad with C1
- Most confusion with other levels.



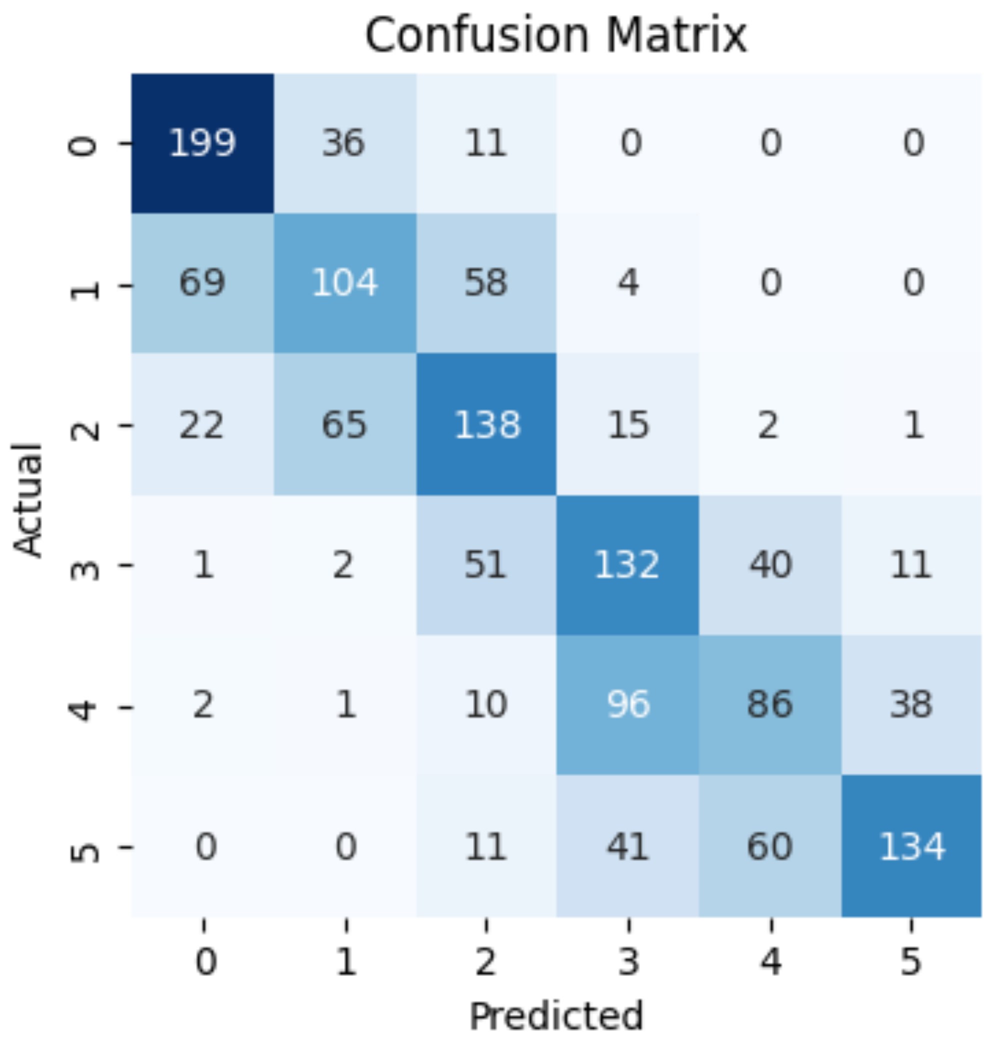
Results (2/4)

- **CamemBert NN**

- Our model achieved an accuracy of 55%

- Strongest at identifying beginner and advanced levels. Quite precise, very few outliers.

- Most confusion with C1 and A2. Most confusion between adjacent levels.



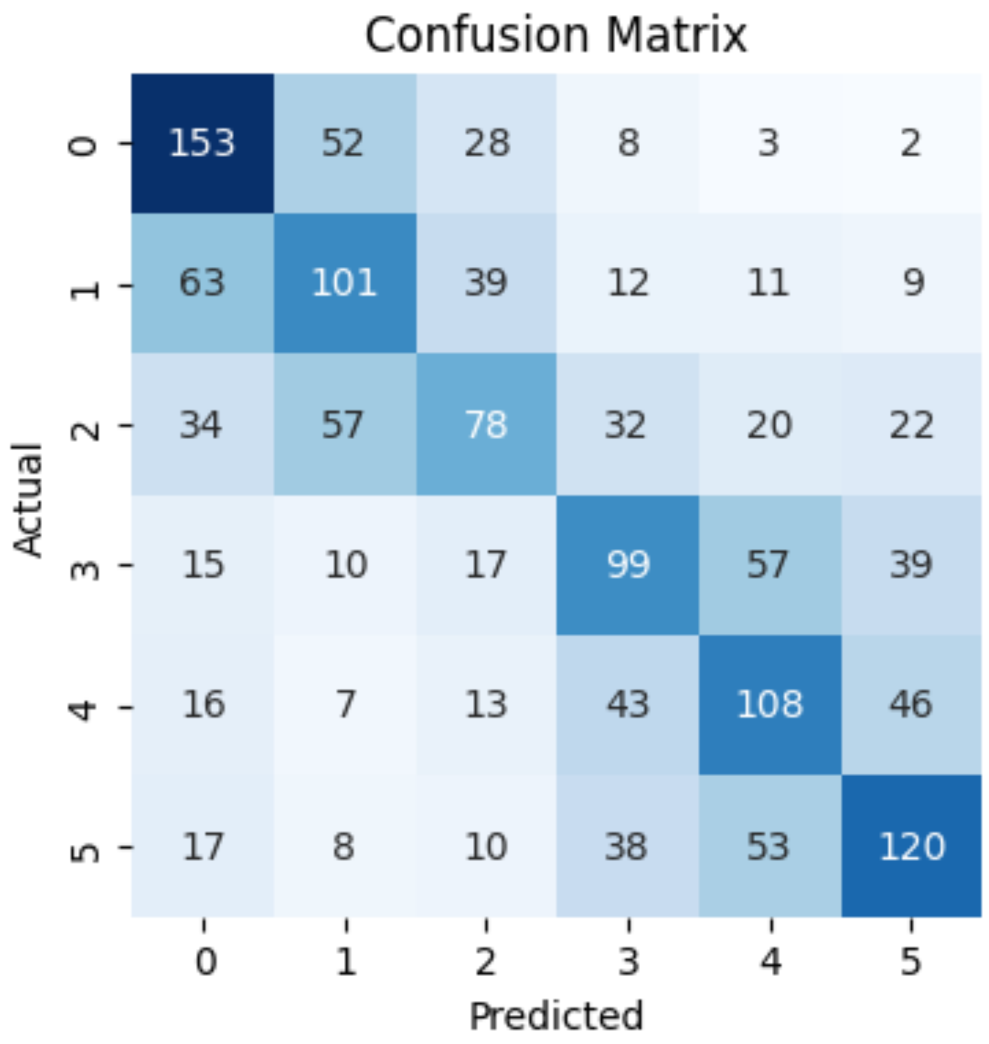
Results (3/4)

- **Word Frequency NN**

- Our model achieved an accuracy of 45.7%

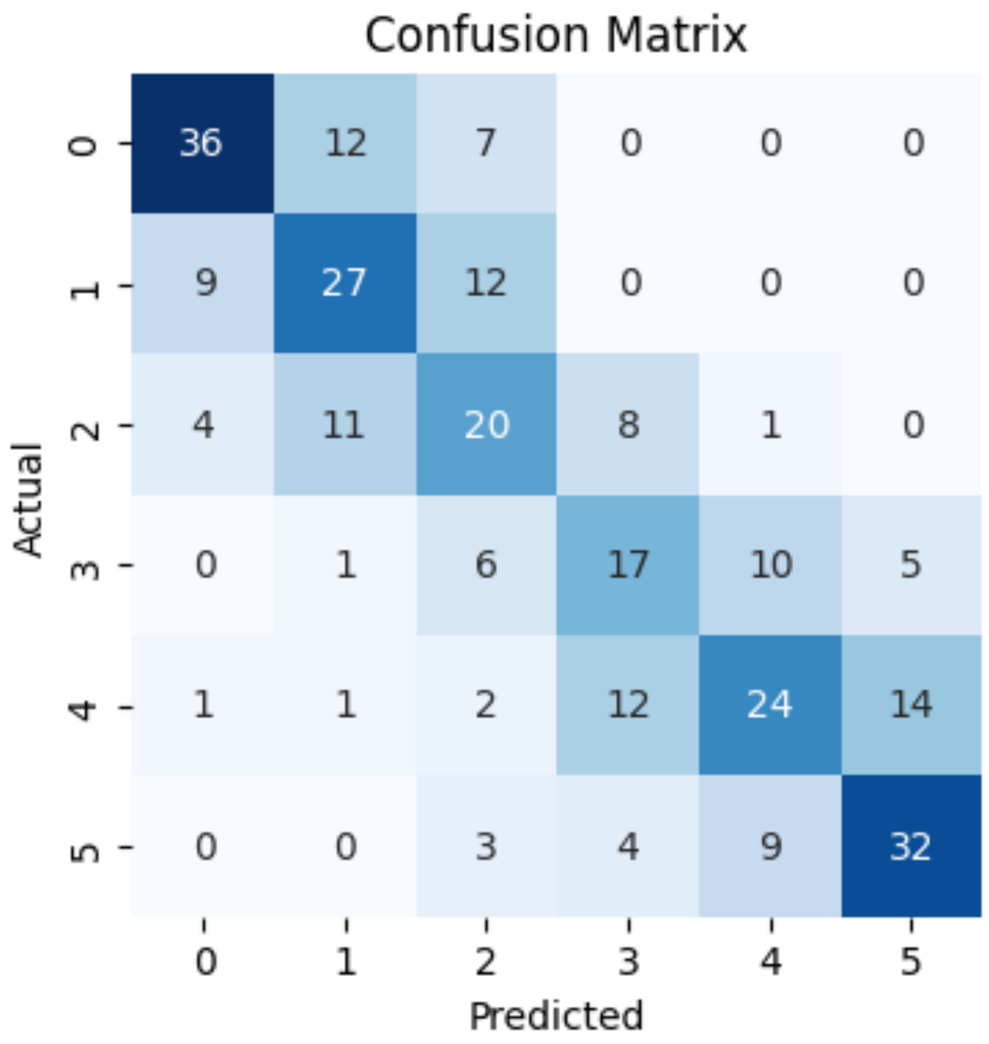
- Strongest at identifying beginner and advanced levels.

- Most confusion between middle levels, such as B1 and B2. A lot of outliers.



Results (4/4)

- **Final NN**
- Our model achieved an accuracy of 54.1%
- Strongest at identifying beginner and advanced levels. Very few outliers
- Most confusion between middle levels such as B1 and B2.



Conclusion: Next Steps in Model Evolution

- Enhanced Differentiation of Intermediate Levels
    - Improve distinction between intermediate levels with advanced feature engineering.

- Real-World Testing
    - Deploy in a beta setting for user feedback and model refinement.

Thank you for listening!