Experiment Plan

1. Data Exploration and Problem Understanding

After exploring the dataset, as well as reading through the literature, I have identified three main challenges for this task:

1. Addressing class imbalance
2. Focusing on recall to avoid missing potential cases
3. Limited resources (100 tests) necessitating high precision

To enhance the dataset, I considered data augmentation techniques. Given more time, I would have used SNOMED CT and UMLS to map HPO terms to other medical terminologies using the xref. This approach would expand the feature set and improve relationships between symptoms, giving more insight into the ontology.

2. Data Preprocessing and Feature Engineering

The preprocessing pipeline will include:

a. Parsing of HPO ontology and creating graph representation
b. Process the HPOA annotations and link to HPO terms
c. Expand patient HPO terms to include ancestor terms (and, in the future, cross-references to other datasets)
d. Implementing techniques to address class imbalance
e. Performing feature selection

3. Model Selection

2 main approaches for modeling:

a) Graph Neural Network (GNN) :
A GNN can restructure data into a graph and capture the hierarchical structure of HPO terms. It is also easy to generalize for other rare or unseen combinations, especially if the model will be used for more than one disease/syndrome. Lastly, GNNs have good inductive bias, which would make them a good choice for symptom relationships.
- Challenges:
    i. More complexity and longer development time
    ii. May be too complex for this classification task

b. Neural Network Variant and Ensemble Method:
Experiment with Feed-Forward Network (FFN) and Attention-based Neural Networks while stacking multiple models to create an ensemble method and determine the best prediction from these models
- Challenges:
    i. Can increase overfitting with limited data
    ii. Harder to debug and maintain with multiple models, also more expensive

Documentation

Preprocessing Approach

1. HPO Ontology: Parsed OBO file, built symptom graph, extracted synonyms and definitions.
2. HPOA Annotations: Mapped diseases to HPO terms, including Pitt Hopkins.
3. Patient Processing: The reported symptoms were gathered and expanded to ancestor terms for hierarchical representation.
4. Feature Engineering:
    a. Weighted Features: Information content is used based on disease annotation frequency.
    b. Text Features: Applied TF-IDF on terms, definitions, and synonyms.
    c. Graph Embeddings: Trained Word2Vec on ontology graph.
5. Feature Combination: Merged different feature types into a single matrix for a multi-modal approach.
6. Target Variable: Created binary indicator for Pitt-Hopkins Syndrome.

I started adding more to my preprocessing pipeline during development. At first, my model only had very basic features and was seeing *horrible* results… The knowledge graph enriched the feature set, which was crucial to model improvement.

During feature engineering, I wanted to connect symptoms to create better/more complex relationships. For example, the “developmental delay” and “hypotonia” features were combined because their co-occurrence was one of the main indicators of Pitt-Hopkins Syndrome.

Model Design and Development

I combined both traditional machine learning and deep learning models to create a stacked ensemble approach comprised of the following:

- Random Forest Classifier
- Gradient Boosting Classifier
- XGBoost Classifier
- Neural Network with custom components
    - Focal loss
    - Dropout
    - Attention mechanism

This ensemble approach helped capture different aspects of the data and find more complex relationships alongside the attention mechanism, which allowed the model to focus only on the most relevant features. Logistic regression was then used to combine these predictions from the base model.

Results and Evaluation

Classification Metrics

Metric	**Value**
Precision	0.0600
Recall	0.8571
F2 Score	0.2344
Accuracy	0.8250

I chose to use F2 over the standard F1 score since it is imperative that potential cases of Pitt Hopkins are not missed. While a test kit may be sent out to someone without it, we don't want to miss any potential patients *with* it in our classification.

Confusion Matrix

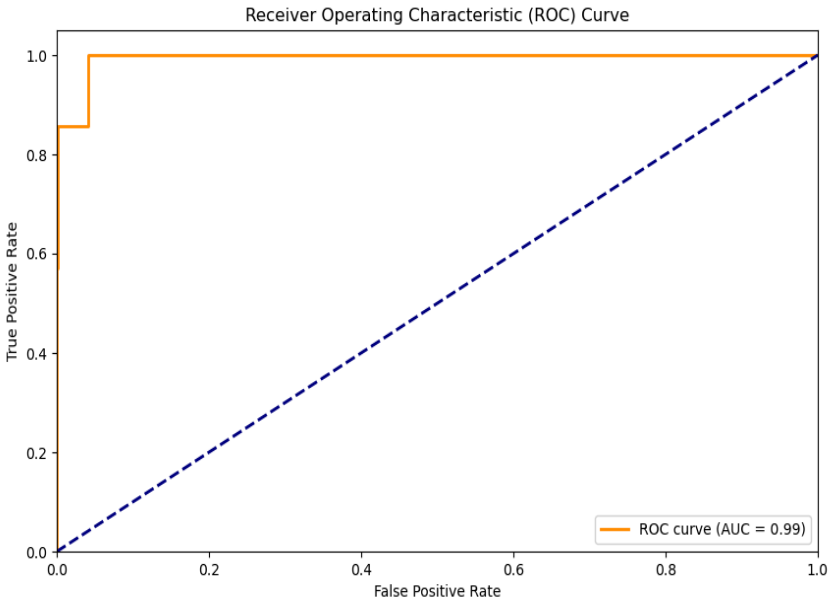
	Predicted Negative	Predicted Positive
Actual Negative	442 (TN)	94 (FP)
Actual Positive	1 (FN)	6 (TP)

Top 10 Important Features

Feature	Importance
lip	0.7832
developmental	0.7704
hypotonia	0.7542
psychomotor	0.7067
tone	0.6381
global	0.5904
muscular	0.5870
milestones	0.5435
walk	0.5268
gait	0.4925

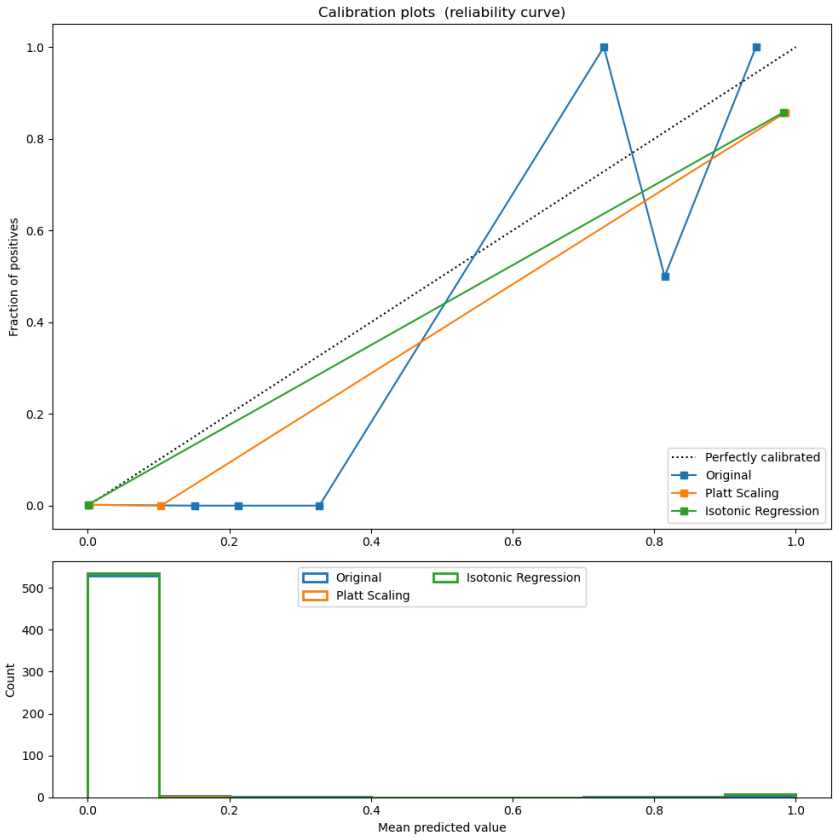
These results align with Pitt Hopkins's main characteristics and may show that additional data or information can help create an even more robust model.

ROC



The ROC curve shows an extremely high AUC of .99, which raises concerns in this context. Namely, the class imbalance creates misleading performance and highlights the model's main issues. Additionally, the rise in the TP rate indicates that this model's performance is very sensitive to the chosen threshold.

Calibration



Looking at the calibration plots, the model's probability estimates are not calibrated well. The original model underestimates the probabilities for lower predicted values and overestimates for higher ones. Using Platt Scaling and Isotonic Regression improved calibration, which is good to know for further experimentation.

Discussion

Overall, the model exhibits high sensitivity in detecting potential Pitt-Hopkins Syndrome cases but suffers from a high false positive rate. In a screening context, this may be an acceptable tradeoff where false positives can be ruled out through subsequent testing. This, however, underscores the need for significant improvements in reducing false positives without impacting the detection of true cases

Addressing class imbalance will be crucial for future development. To better represent the minority class, I could experiment with resampling techniques like ADASYN or anomaly detection. For model improvement, developing a custom loss function could more accurately reflect the cost of misclassification in the medical context. Lastly, with more time, I would have explored using a GNN or a different ensemble structure to test what performed best alongside the other factors of resampling, feature selection methods, etc.

Productionalization Plan

1. Deployment environment: Use a HIPAA-compliant cloud environment to ensure security and compliance
2. User Interface: Use an interface for patient inputs of symptom data using standardized HPO terms
3. Implement real-time processing of the input data and use the preprocessed knowledge graph for symptom relationships
4. Deploy the model to process inputs and generate probability scores for Pitt-Hopkins Syndrome (or other disease) likelihood
5. Create a custom thresholding system to recommend testing for top candidates (within 100 test constraints)

Taking this from production to development would require containerizing the model using Docker so that the environment is consistent and deployed on a Kubernetes cluster for scalability and management. For model serving, we can use a Flask API to get prediction and model information endpoints. For lifecycle management, MLflow could be used for experiment tracker, versioning, etc., which allows for model version comparing and rollbacks.