Part a: Predictive models for our annotated data

1. Started with ***OrdinalClassfier*** *with Bow logistic regression a*s our dataset is multi-class with 4 classes:
   "Child", "Adolescent", "Young Adult", "Adult"
2. Changed ***binary_bow_featurize*** method by adding additional features ***'adult_feat'*** and ***'child_feat'*** with a list of adult and children words.

Test accuracy for improved bag of word features model: 0.480, 95% CIs: [0.382 0.578]

3. Tried a second model, ***"word2vec-google-news-300",*** a pre-trained word embedding model developed using the Word2Vec algorithm. We used the open-source Python NLP library ***gensim***. ***OrdinalClassifierWithEmbeddings*** classifier derives from OrdinalClassifier with overridden process method and uses this model in its featurize method ***word_embedding_featurize.***
Source: ***https://radimrehurek.com/gensim/models/word2vec.html, stackoverflow***

Test accuracy for model with GloVe embeddings: 0.430, 95% CIs: [0.333 0.527]
The difference in accuracy was minuscule.

4. Logistic Regression - After an initial run, we noticed that a lot of the top features for each class were irrelevant words such as 'the', 'of', 'and', etc. So, we first cleaned our data so that each text would not include these words. Additionally, we implemented TF-IDF to similarly handle common words across documents, but also emphasize important words for each class.
Test accuracy for best dev model: 0.515, 95% CIs: [0.417 0.614]

5. BERT - on top of the initial BERT model, we used fine tuning techniques to improve the performance of our model. For example, we utilized a dropout layer to prevent overfitting in our model and develop a more robust representation that better generalizes our data. We also used a scheduler to adjust the learning rate automatically during training to help the model converge more efficiently.
Test accuracy for best dev model: 0.440, 95% CIs: [0.343 0.537]

6. ELMo (Embeddings from Language Models) was used for an attempt to improve accuracy but we ran into issues with calculating the test accuracy for the whole dataset. Excluding stopwords , the embeddings would then capture the meaning of words based on their context within the text. ELMo uses surrounding words when creating word representations using a large corpora of text data. By Leverage transfer learning, we would use the pre-trained embeddings as feature representation for ordinal rating predictions.

In conclusion, the limited improvement in accuracy can be attributed to the relatively small dataset size. We opted for multi-class annotation instead of binary, which necessitates a larger dataset to achieve significantly higher accuracy.

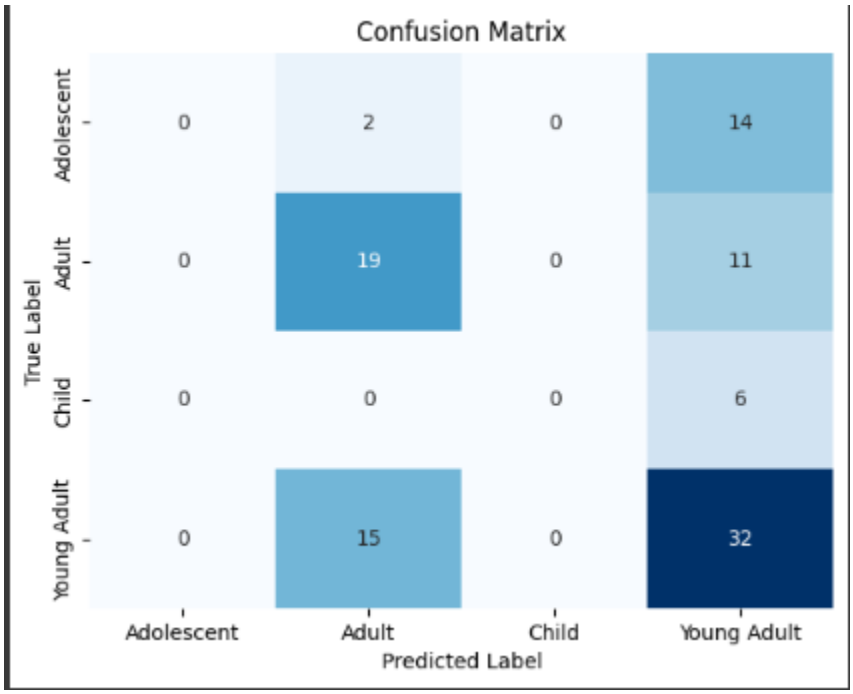
Part 2: Analysis

1. Does your model learn features of the phenomenon that you didn't consider in your guidelines that might cause you to rethink the category boundaries?

Yes, somewhat. it's the edges of the categories that become blurred and cross over easily. Our initial guidelines for the categories are pretty solid but the textual patterns are very distinctive in most of the data and this phenomenon is learnt by the model.

The GloVe embedding model exhibits a similar phenomenon. It represents each word in the input text with a GloVe embedding vector, capturing the semantic meaning of words within the data context. These embedding vectors then serve as input features for the EmbeddingsOrdinalClassifier.

2. What labels are often mistaken for each other? (e.g., using a confusion matrix (https://scikitlearn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html))



Here is an example of a confusion matrix from our Logistic Regression model. This is based on the test set. As we can see, there is a major problem with our model predicting both Adolescent and Child labels. This could be due to a lack of representation in the training set or an overall high similarity across all the text classes that makes it hard to differentiate between them. Additionally, we see that the categories 'Adult' and 'Young Adult' were getting confused with each other due to blurred criteria and textual patterns. Overall, this gives insight into the incorrect labeling (Adult vs. Young Adult), however depicts an anomaly present within this model which we can attribute to a small dataset size and possibly overfitting.

3. What features are learned to most define the classes?

- Basic features are the token word features.
- ‘adult_feat’ for adult rating words and ‘child_feat’ for common words in children’s books.
- For OrdinalClassifierWith Embeddings, GloVe embeddings capture semantic relationships between words and are used as features.

```
Class 'Adolescent' top features:
ago: 0.351
evening: 0.281
off: 0.280
early: 0.276
familiar: 0.273
warned: 0.265
standing: 0.263
opened: 0.260
observations: 0.259
until: 0.257

Class 'Adult' top features:
times: 0.369
before: 0.349
between: 0.330
died: 0.323
sight: 0.298
present: 0.298
able: 0.290
): 0.289
road: 0.288
hope: 0.277

Class 'Child' top features:
children: 0.610
visit: 0.364
jolly: 0.355
adventures: 0.354
why: 0.349
surprise: 0.340
make: 0.308
nice: 0.300
asked: 0.300
farmer: 0.294

Class 'Young Adult' top features:
shows: 0.347
london: 0.343
master: 0.324
young: 0.319
some: 0.310
society: 0.290
government: 0.287
same: 0.282
visits: 0.264
ships: 0.263
```

- From the Logistic Regression model, we observe results that align closely with expectations for the top features associated with each class. For instance, the word "children" strongly correlates with the 'Child' class, which intuitively makes sense given its frequent use in children's literature. Moreover, other significant features for the 'Child' class include terms like 'nice', 'jolly', 'adventures', and 'surprise', which are commonly found in narratives aimed at younger readers. In contrast, the 'Young Adult' class features more sophisticated terms such as 'society' and 'government', reflecting themes typically absent in children's and adolescent literature. Overall, there still remain some questionable top features for each class, however this can most likely be attributed to the small test set that we have.

4. What kind of systematic mistakes does your model make?

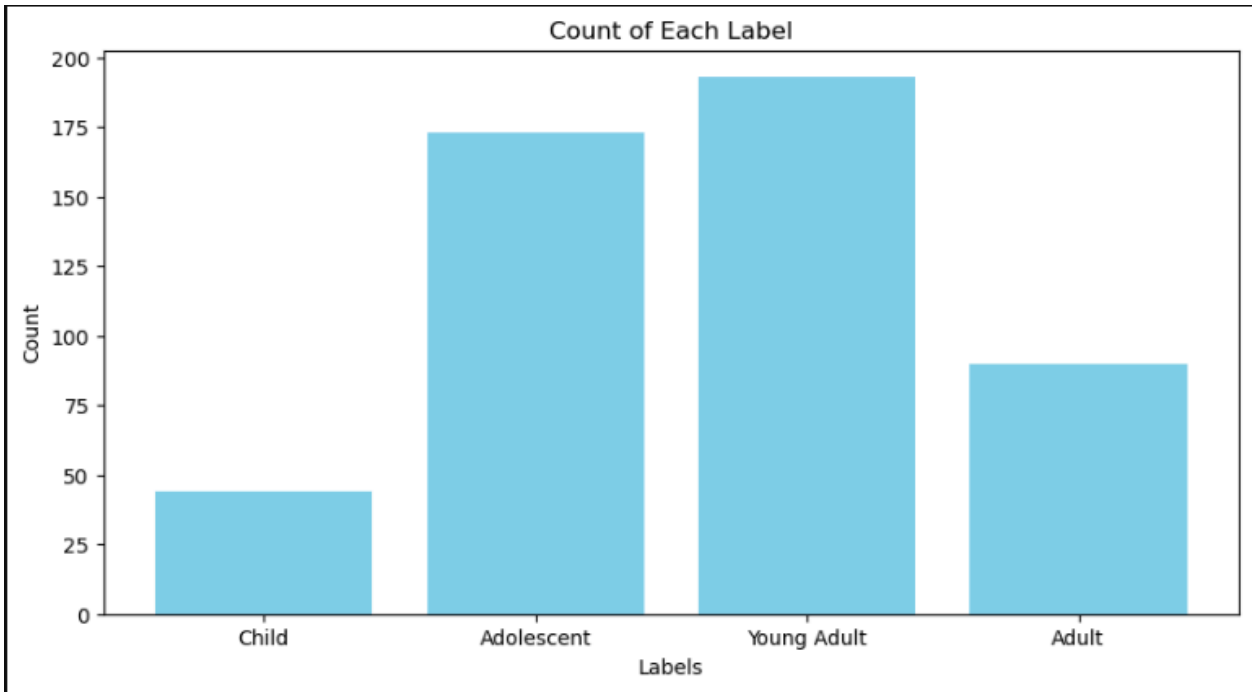
Our models consistently predict categories that are adjacent to the true category but not the exact correct category. For example, if the true category is "Young Adult", the model consistently predicts "Adult" or "Adolescent" instead of the correct category. This is due to the complexity of our task and nuances that

go into reading levels. We had similar inconsistencies when we compared our own annotations, so it makes sense as to why a model would perform similarly.

5. Are there any biases your model makes? *There is no language bias as this only uses English. However, our models will have representation bias due to some classes having less representation than others in our dataset.*

6. Think about the level of balance in your dataset: Is one label extremely prevalent? How could this impact the model you developed? Is your dataset a good candidate for strategies like oversampling?

Yes. There is a fair amount of class imbalance in our dataset. There are 4 classes in our dataset. 'Adult' and 'Child' classes are not as prevalent as 'Adolescent' and 'Young Adult'. 'Adult' and 'Child' classes are underrepresented with respect to the other 2 classes. So they are naturally a good candidate for oversampling.



7. Discussion/Reflection on the human annotation process

As we explore the performance of our various models, it is important to remember the difficulties that we had while annotating. Within our annotation process, we developed very strict and robust guidelines, however we still continued to see differences across annotators. Even with strict guidelines, our classification task can be highly subjective, particularly for texts that do not clearly fall into one category. Different annotators may interpret texts differently, especially for texts that straddle the line between two categories, such as 'Adolescent' and 'Young Adult'. This subjectivity inevitably introduces variability and noise into the labeled data, which can confuse the model during training and lead to inconsistent predictions. Additionally, language use can overlap significantly between categories,

especially in age-related classification where stages may share common characteristics. For instance, vocabulary, themes, and maturity levels in 'Adolescent' texts might closely resemble those in 'Young Adult' texts. If human annotators struggle to distinguish these, it's likely that models, which rely on patterns in the data, will encounter similar confusion. Overall, although we sometimes encountered poor accuracy in our models, it is relevant to remember the subjectivity and difficulty that went into our initial annotation task.

8. Weaknesses / Shortcomings / Future work

Apart from examination of the human annotation process, the largest weakness of all of our models came from our small dataset. With a test set of just 300 entries, combined with an already complex and nuanced task, it was not surprising to see our accuracy scores hovering around 0.4 - 0.5. However, all models were significantly improved in comparison to baseline models, so it is promising that with a larger dataset, we would be able to finetune and edit these models to achieve even greater performance metrics.

For future work, the main focus would definitely be to create a larger dataset. Additionally, it may be helpful to revisit annotation guidelines to assess possible the removal/addition of new labels in order to handle the common mistakes between labels (e.g. classifying 'Adult' vs. 'Young Adult')