Complexity of literary passages - Kaggle competition.

Capstone proposal - Udacity -Machine Learning Engineer Nanodegree.

Domain Background

“Can machine learning identify the appropriate reading level of a passage of text, and help inspire learning ? Reading is an essential skill for academic success. When students have access to engaging passages offering the right level of challenge, they naturally develop reading skills.”

- Quoting Kaggle CommonLit Readability competition.

The main idea here is to take leverage on the NLP knowledge the Nanodegree have gave me and find a way to tackle this real world problem. My thoughts right now are centered on using classic indicators of read complexity and some novelty ML techniques to give better predictions.

Problem Statement

“Currently, most educational texts are matched to readers using traditional readability methods or commercially available formulas. However, each has its issues. Tools like Flesch-Kincaid Grade Level are based on weak proxies of text decoding (i.e., characters or syllables per word) and syntactic complexity (i.e., number or words per sentence). As a result, they lack construct and theoretical validity. At the same time, commercially available formulas, such as Lexile, can be cost-prohibitive, lack suitable validation studies, and suffer from transparency issues when the formula’s features aren’t publicly available.”

- Quoting Kaggle CommonLit Readability competition.

I see a lot of potential with the idea, imagine yourself for a moment, if you could know beforehand buying a book the complexity of the text that lurks inside, that may encourage you to find books that fit your comfort level. I could easily imagine every book in a shelve with an stamp like the “best seller” one but stating the complexity. Then buying books for kids will be easier as well as for foreign people to pick books written in other languages also for older people with cognitive problems, the possibilities are endless.

Since the current indexes for text complexity aren’t perfect, nor accurate with real people opinions, there may be a way to improve current result with the help of NLP and ML. Finally making a humble contribution to cause greater than myself as finding a free, open-source complexity index is a reward on itself.

Jose Nuñez

Datasets and Inputs

The dataset that Kaggle provided takes this form:

Training data

```
print(all_df.shape, '\n')
all_df.head()
```

```
(2834, 6)
```

	id	url_legal	license	excerpt	target	standard_error
0	c12129c31	NaN	NaN	When the young people returned to the ballroom, it presented a decidedly changed appearance. Instead of an interior scene, it was a winter landscape.\nThe floor was covered with snow-white canvas, not laid on smoothly, but rumpled over bumps and hillocks, like a real snow field. The numerous palms and evergreens that had decorated the room, were powdered with flour and strewn with tufts of cotton, like snow. Also diamond dust had been lightly sprinkled on them, and glittering crystal icicles hung from the branches.\nAt each end of the room, on the wall, hung a beautiful bear-skin rug.\nThese rugs were for prizes, one for the girls and one for the boys. And this was the game.\nThe girls were gathered at one end of the room and the boys at the other, and one end was called the North Pole, and the other the South Pole. Each player was given a small flag which they were to plant on reaching the Pole.\nThis would have been an easy matter, but each traveller was obliged to wear snowshoes.	-0.340259	0.464009
1	85aa80a4c	NaN	NaN	All through dinner time, Mrs. Fayre was somewhat silent, her eyes resting on Dolly with a wistful, uncertain expression. She wanted to give the child the pleasure she craved, but she had hard work to bring herself to the point of overcoming her own objections.\nAt last, however, when the meal was nearly over, she smiled at her little daughter, and said, "All right, Dolly, you may go."\n"Oh, mother!" Dolly cried, overwhelmed with sudden delight. "Really?\nOh, I am so glad! Are you sure you're willing?"\n"I've persuaded myself to be willing, against my will," returned Mrs. Fayre, whimsically. "I confess I just hate to have you go, but I can't bear to deprive you of the pleasure trip. And, as you say, it would also keep Dotty at home, and so, altogether, I think I shall have to give in."\n"Oh, you angel mother! You blessed lady! How good you are!" And Dolly flew around the table and gave her mother a hug that nearly suffocated her.	-0.315372	0.480805
2	b69ac6792	NaN	NaN	As Roger had predicted, the snow departed as quickly as it came, and two days after their sleigh ride there was scarcely a vestige of white on the ground. Tennis was again possible and a great game was in progress on the court at Pine Laurel. Patty and Roger were playing against Elise and Sam Blaney, and the pairs were well matched.\nBut the long-contested victory finally went against Patty, and she laughingly accepted defeat.\n"Only because Patty's not quite back on her game yet," Roger defended; "this child has been on the sick list, you know, Sam, and she isn't up to her own mark."\n"Well, I like that!" cried Patty; "suppose you bear half the blame, Roger. You see, Mr. Blaney, he is so absorbed in his own Love Game, he can't play with his old-time skill."\n"All right, Patsy, let it go at that. And it's so, too. I suddenly remembered something Mona told me to tell you, and it affected my service."	-0.580118	0.476676
3	dd1000b26	NaN	NaN	And outside before the palace a great garden was walled round, filled full of stately fruit-trees, gray olives and sweet figs, and pomegranates, pears, and apples, which bore the whole year round. For the rich south-west wind fed them, till pear grew ripe on pear, fig on fig, and grape on grape, all the winter and the spring. And at the farther end gay flower-beds bloomed through all seasons of the year; and two fair fountains rose, and ran, one through the garden grounds, and one beneath the palace gate, to water all the town. Such noble gifts the heavens had given to Alcinous the wise.\nSo they went in, and saw him sitting, like Poseidon, on his throne, with his golden sceptre by him, in garments stiff with gold, and in his hand a sculptured goblet, as he pledged the merchant kings; and beside him stood Arete, his wise and lovely queen, and leaned against a pillar as she spun her golden threads.	-1.054013	0.450007
4	37c1b32fb	NaN	NaN	Once upon a time there were Three Bears who lived together in a house of their own in a wood. One of them was a Little, Small, Wee Bear; and one was a Middle-sized Bear, and the other was a Great, Huge Bear. They had each a pot for their porridge; a little pot for the Little, Small, Wee Bear; and a middle-sized pot for the Middle Bear; and a great pot for the Great, Huge Bear. And they had each a chair to sit in; a little chair for the Little, Small, Wee Bear; and a middle-sized chair for the Middle Bear; and a great chair for the Great, Huge Bear. And they had each a bed to sleep in; a little bed for the Little, Small, Wee Bear; and a middle-sized bed for the Middle Bear; and a great bed for the Great, Huge Bear.	0.247197	0.510845

It has 6 columns, and 2834 rows. The columns are: id, url_legal, license, excerpt, target and standard_error. The id is just the excerpt identifier, both url_legal and license contains info about the excerpt license of use. The excerpt is the actual text you want to find out the complexity, the target columns is the average reading ease according to a panel of teachers, and the standard_error is the error of such average. Please notice the target variable is a continuous number, this means this is regression problem.

They also include a test dataset, but this ones doesn't have neither the target nor the standard_error. So, we must train the model solely using the excerpt text.

In addition to the above dataset I made some research and found an open source repository which contains some info regarding the frequency of English words, specifically for the top 50.000 more frequent words. My idea is to use this auxiliar dataset as well to determine the "rarity" of the words in the excerpt. The link to this repo is here: https://github.com/hermitdave/FrequencyWords.

Solution Statement

The solution is to create a model that accurately predicts the complexity of a text and it needs to be better than classic indexes (with a lesser RMSE). Then host the model into a web application that let's people know the complexity of an excerpt in a matter of seconds. So not only accuracy maters but time as well.

Benchmark Model

I will use two models as benchmark, first a basic linear regression between the Flesch-Kincaid test index and the target variable, this is the most popular complexity index so performing better that it, its an accomplishment in on itself. This model will give us an idea of how good are classic indicators at estimating the real complexity of a excerpt.

Secondly I'll use other Kaggle contestant RMSE to compare against mine.

Evaluation Metrics

The benchmark for the model will be the root mean square error (RMSE) using the predictions of the model and the real values as inputs to the formula.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Project Design

Right now I haven't design a full solution, I'm still exploring ideas to tackle this complicated problem and so far I see 3 possible candidates:

* Perform some feature engineer process on the excerpt to describe it better, and use such quantitative characteristic as input for a traditional regression model as a support vector regressor. We can create variables from the excerpt such as the Flesch-Kincaid test index, calculate the average number of words per sentence, the average rarity of the words in the excerpt (using the auxiliar dataset), counting the number of unique words in the text, you name it. Then use those descriptive number to train a support vector regression as usual.

* Train a LSTM in the pure style of the sentiment analysis mini-project we did in the first part of the NanoDegree. This implies creating a word dictionary and then transform every excerpt into a vector using such vocabulary. Then feed this vectors to the net and hope it will learn the complexity from it. Also notice that in the mini project we did a classification with only 2 possible outcomes: positive or negative review. In this case we have a continuous variable as output, so we may change the last layer of the net to predict a number instead of a class.

* Use a state of the art pre-trained model such as Google's Transformer BERT to accomplish the regression task. A lot of participants are suggesting this is the approach that provides the best results and as a Machine Learning Engineer student I should be able to learn new concepts and apply them. so probably I'll try this as my last resort if I don't get nice result for my other two alternatives.

Also bear in mind that 2834 records to train a net is a quite low number of examples, so I'm considering to perform some form of data augmentation process, maybe duplicating the original excerpts and replacing some words with synonymous words. I still need to research more on how to do this correctly.