Feed the Machine: Self-Assessment Scoring

Project 3

June 1, 2021

Susan Farago, David Jimenez, Austin Olea, Catherine Poirier, Jenni Davis & Elizabeth Conway

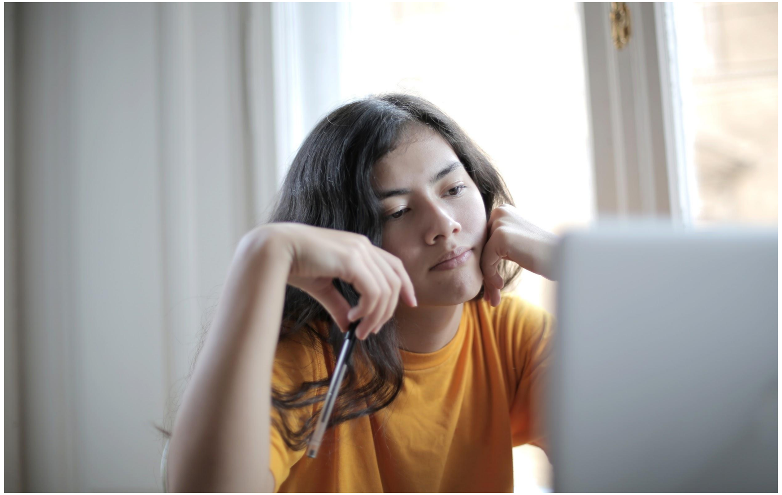
Once Upon a Time.........

Problem Statement:

Once upon a time, Epi, spent her days creating a self assessment questionnaire for trainers to complete after the conclusion of a course on teaching in a remote environment.

Epi's goal was to find a way to measure the responses & provide insight on what the trainers learned.

She asked, 'can machine learning accurately and effectively evaluate a trainer's response to a collection of essay-style questions & predict the trainer's training and facilitation skills?'



Source Data:

File #1

- Extracted from SalesForce & cleaned utilizing Tableau.
- Essay responses from 2,388 trainers to twenty open-ending questions from June 2020 - May 2021, resulting in 42,998 rows of data.

File #2

- Extracted from SalesForce & cleaned utilizing Tableau.
- Student scoring on the respective trainer’s training and facilitation skills.
- 912 trainers taught courses of those 486 received scoring.
- Final cleaned data included 312 trainers and 6,240 records.

Tools Used:

Cleaning the Data:

- Tableau

Preprocessing:

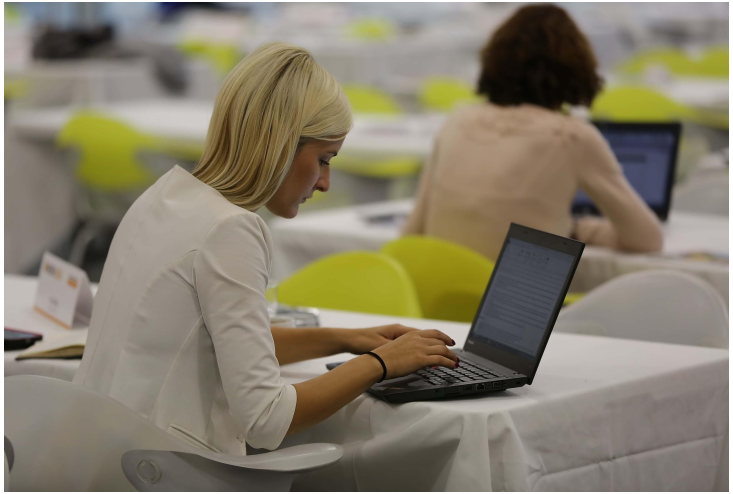
- Python Pandas
- Python NLTK
- Matplotlib

Machine Learning:

- Scikit-learn
  - Naive Bayes Classifier
  - GaussianNB
- Linear Regression
- Random Forest Regressor

Showing the Work:

- HTML / CSS
- GitHub Pages



Analysis: Preprocessing Steps

Phase #1: Preprocessing Steps

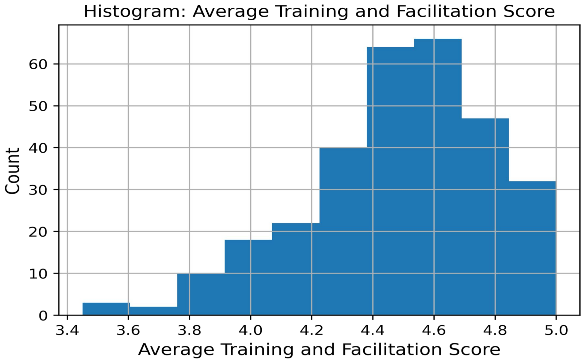
1. Tokenization
2. StopWords
3. Lemmatization
4. POS Tag
5. TF-IDF Vectorizer

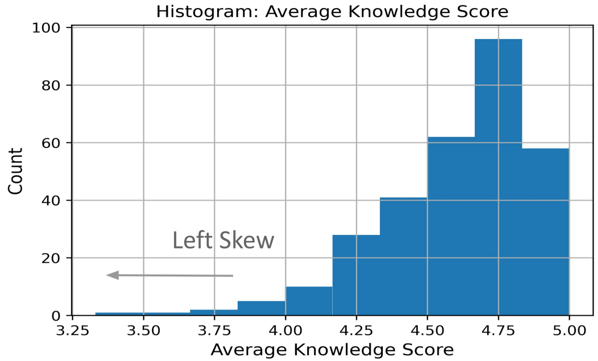
Phase #2: Data Model/Machine Learning

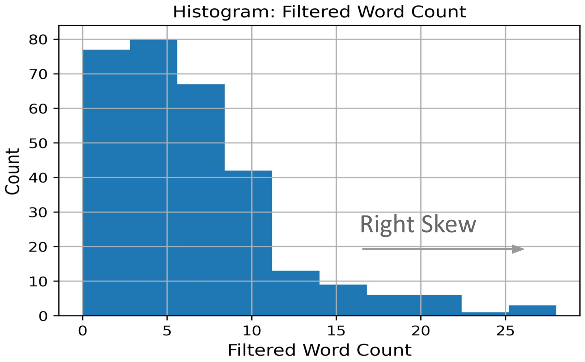
6. Vader Sentiment
7. Linear Regression
8. Random Forest Regressor

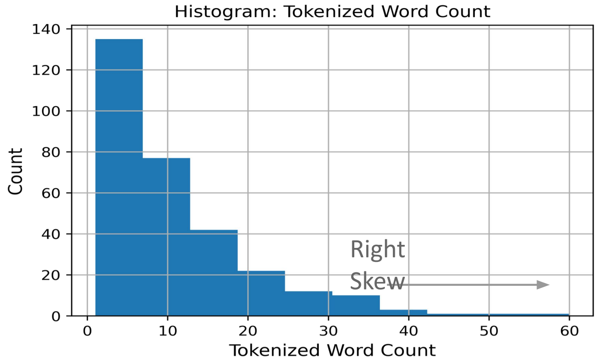
Squeeze the Data (EDA)

Data Distributions:


Histogram: Average Training and Facilitation Score


Histogram: Average Knowledge Score


Histogram: Filtered Word Count


Histogram: Tokenized Word Count

Regression Analysis Questions:

Question #11: What is the one tip that you would share with someone preparing to teach a virtual class for the first time?

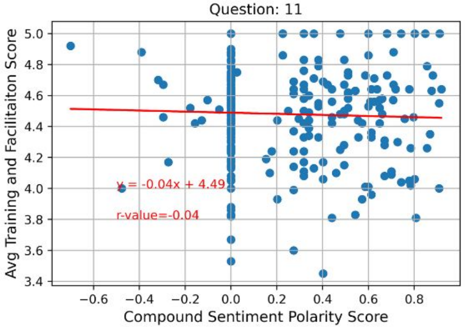
Question #12: What would you recommend as a tip or method to keep remote learners engaged and why?

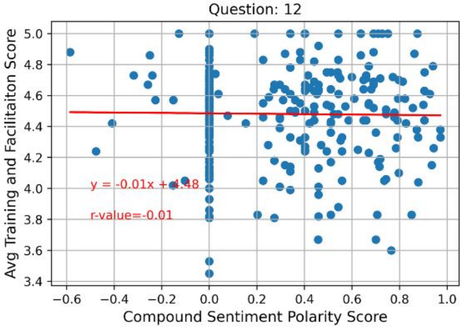
Question #13: What is one activity your co-trainer can do to support you while you are teaching a remote class?

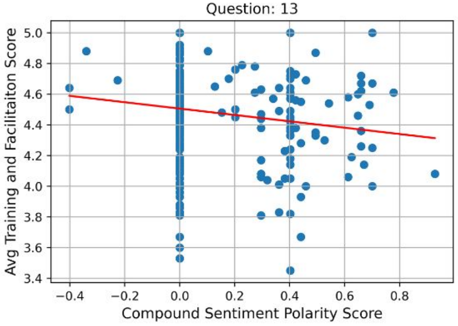
Question #14: What is one way you can gain empathy for the student's virtual classroom experience?

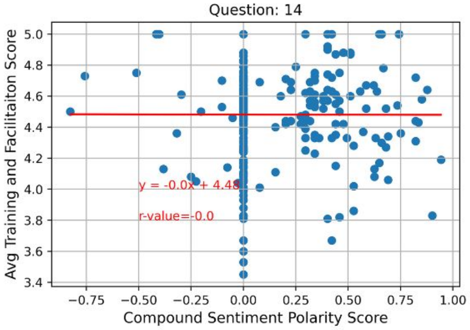
Question #15: What do you think is one benefit of having at least two trainers for your class?

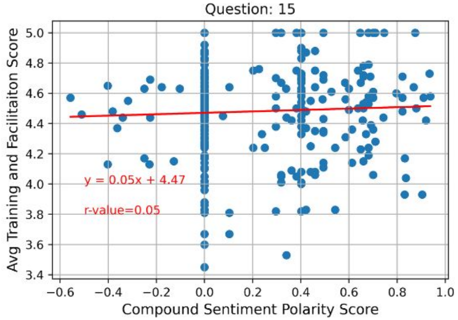
Regression Analysis: 11- 15


Question: 11

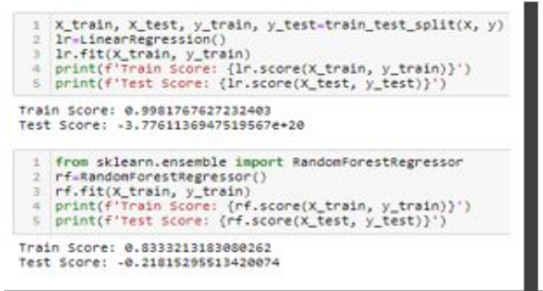

Question: 12


Question: 13


Question: 14


Question: 15

ML - Questions 11 – 15:

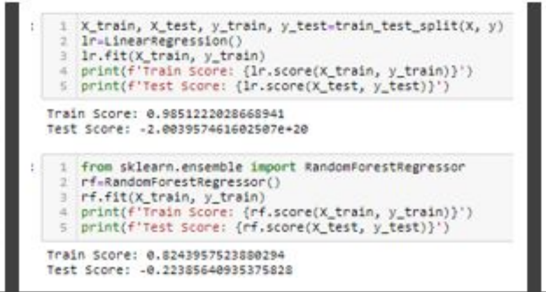


```python
X_train, X_test, y_train, y_test=train_test_split(X, y)
lr=LinearRegression()
lr.fit(X_train, y_train)
print(f'Train Score: {lr.score(X_train, y_train)}')
print(f'Test Score: {lr.score(X_test, y_test)}')
```

```
Train Score: 0.9981767627232403
Test Score: -3.7761136947519567e+20
```

```python
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train, y_train)
print(f'Train Score: {rf.score(X_train, y_train)}')
print(f'Test Score: {rf.score(X_test, y_test)}')
```

```
Train Score: 0.8333213183080262
Test Score: -0.21815295513420074
```

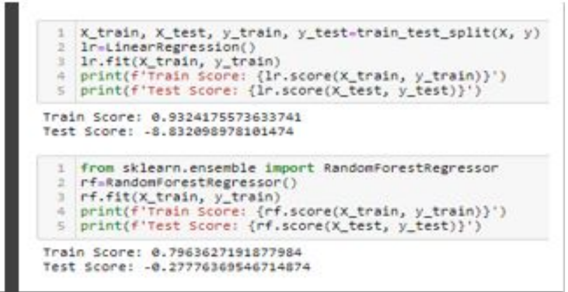


```python
X_train, X_test, y_train, y_test=train_test_split(X, y)
lr=LinearRegression()
lr.fit(X_train, y_train)
print(f'Train Score: {lr.score(X_train, y_train)}')
print(f'Test Score: {lr.score(X_test, y_test)}')
```

```
Train Score: 0.9851222028668941
Test Score: -2.003957461602507e+20
```

```python
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train, y_train)
print(f'Train Score: {rf.score(X_train, y_train)}')
print(f'Test Score: {rf.score(X_test, y_test)}')
```

```
Train Score: 0.8243957523880294
Test Score: -0.22385640935375828
```

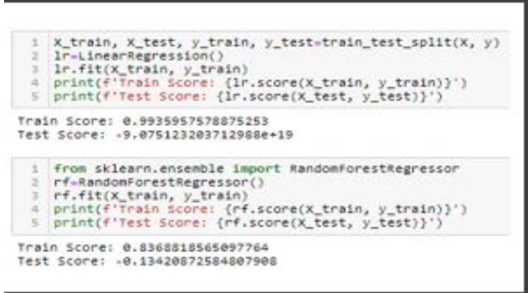


```python
X_train, X_test, y_train, y_test=train_test_split(X, y)
lr=LinearRegression()
lr.fit(X_train, y_train)
print(f'Train Score: {lr.score(X_train, y_train)}')
print(f'Test Score: {lr.score(X_test, y_test)}')
```

```
Train Score: 0.9324175573633741
Test Score: -8.832098978101474
```

```python
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train, y_train)
print(f'Train Score: {rf.score(X_train, y_train)}')
print(f'Test Score: {rf.score(X_test, y_test)}')
```

```
Train Score: 0.7963627191877984
Test Score: -0.27776369546714874
```

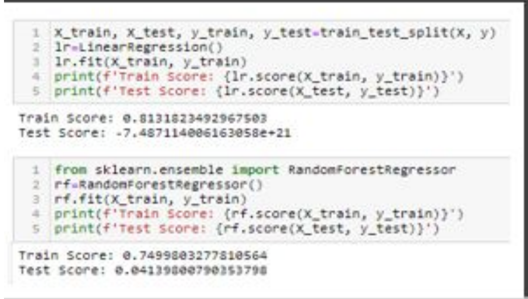


```python
X_train, X_test, y_train, y_test=train_test_split(X, y)
lr=LinearRegression()
lr.fit(X_train, y_train)
print(f'Train Score: {lr.score(X_train, y_train)}')
print(f'Test Score: {lr.score(X_test, y_test)}')
```

```
Train Score: 0.9935957578875253
Test Score: -9.075123203712988e+19
```

```python
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train, y_train)
print(f'Train Score: {rf.score(X_train, y_train)}')
print(f'Test Score: {rf.score(X_test, y_test)}')
```

```
Train Score: 0.8368818565097764
Test Score: -0.13420872584807908
```



```python
X_train, X_test, y_train, y_test=train_test_split(X, y)
lr=LinearRegression()
lr.fit(X_train, y_train)
print(f'Train Score: {lr.score(X_train, y_train)}')
print(f'Test Score: {lr.score(X_test, y_test)}')
```

```
Train Score: 0.8131823492967503
Test Score: -7.487114006163058e+21
```

```python
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor()
rf.fit(X_train, y_train)
print(f'Train Score: {rf.score(X_train, y_train)}')
print(f'Test Score: {rf.score(X_test, y_test)}')
```

```
Train Score: 0.7499803277810564
Test Score: 0.04139800790353798
```

Showing the Work

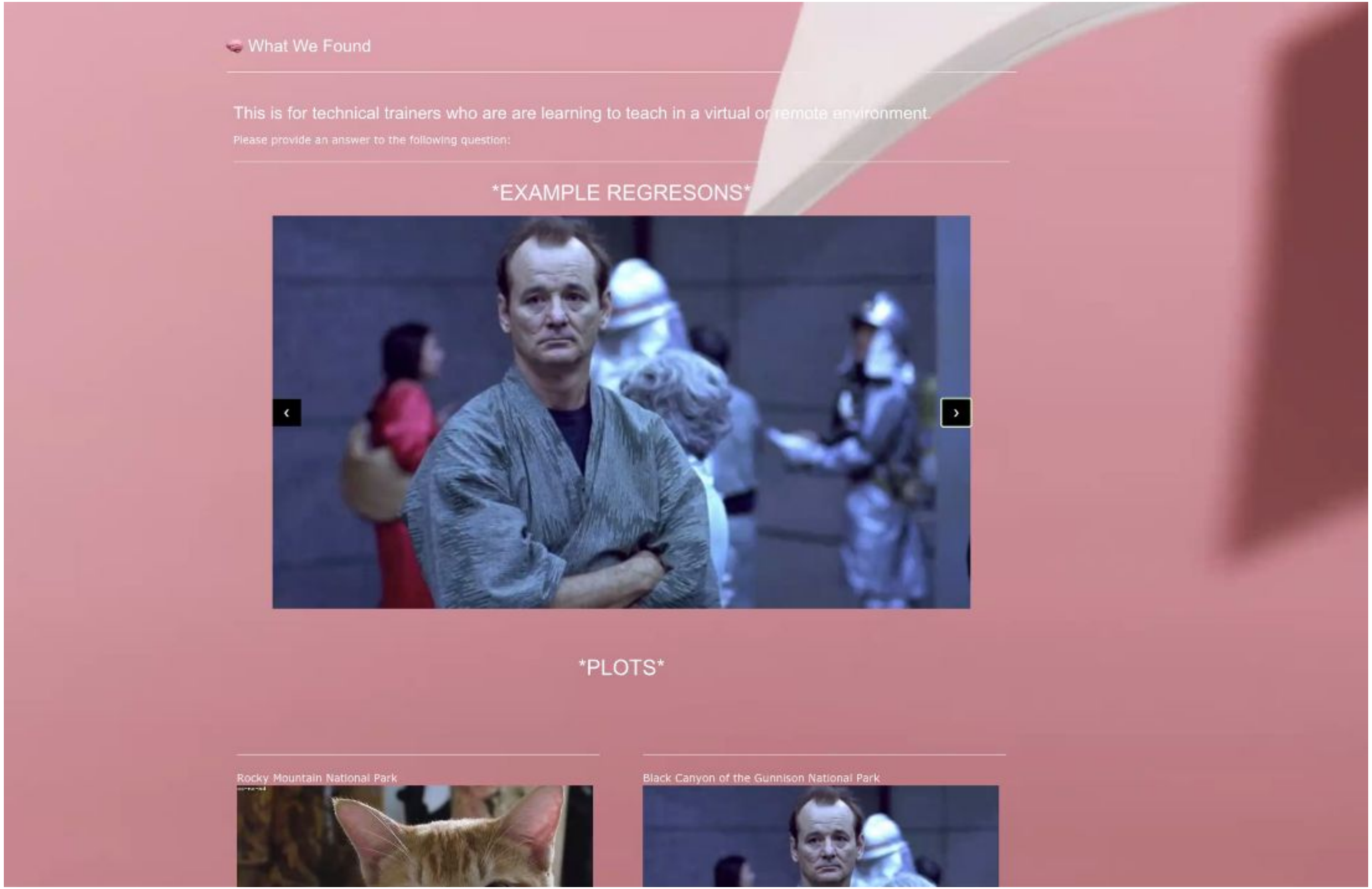
Project Results Page:

User Friendly Data Interface:

- HTML
- CSS
- Java
- Script SCSS

Presents Results of “Feed The Machine Class Project”

```
<div class="container">
    <div class="row">
        <div class="col-md-6 col-lg-8"> <!-- __________ BEGIN NAVIGATION
            <nav id="nav" role="navigation">
                <ul>
                    <li><a href="index.html">Home</a></li>
                    <li><a href="home-events.html">Home Events</a></li>
                    <li><a href="multi-col-menu.html">Multiple Column Menu
                    <li class="has-children"> <a href="#" class="current">
                        <ul>
                            <li><a href="tall-button-header.html">Tall But
                            <li><a href="image-logo.html">Image Logo</a></
                            <li class="active"><a href="tall-logo.html">Ta
                        </ul>
                    </li>
                    <li class="has-children"> <a href="#">Carousels</a>
                        <ul>
                            <li><a href="variable-width-slider.html">Variab
```



What We Found

This is for technical trainers who are are learning to teach in a virtual or remote environment.

Please provide an answer to the following question:

EXAMPLE REGRESONS

‹ ›

PLOTS

Rocky Mountain National Park

Black Canyon of the Gunnison National Park

Conclusion

Conclusion:

Epi discovered that she was unable to predict a trainer’s facilitation and training skills based on the responses to the essay style questions.

Although disappointed, an energized Epi went back to the drawing board to develop questions that would produce meaningful and predictable data.

Real-world Application:

Not all questions have the same intent, purpose, or outcomes: self assessment vs performance vs multiple choice.

Sentiment analysis will be used to assess questions to get a sense of introducing bias in questions.

Machine learning models will be used to test predictability and efficacy of various exam questions and types.

If we can measure effective exam questions, can we ask fewer questions with the same or better reliability.

SUCCESS!

Massive thank you to Jenni, David, Elizabeth, Austin, and Catherine for helping me find answers to these questions and give me an impressive tool I will use in my day-to-day job.

You are awesome! (97.8% positive)

I could not have done this without you! (90.4% positive)