This notebook is an exercise in the AI Ethics course. You can reference the tutorial at this link.

In the tutorial, you learned about six different types of bias. In this exercise, you'll train a model with **real data** and get practice with identifying bias. Don't worry if you're new to coding: you'll still be able to complete the exercise!

Introduction

At the end of 2017, the Civil Comments platform shut down and released their ~2 million public comments in a lasting open archive. Jigsaw sponsored this effort and helped to comprehensively annotate the data. In 2019, Kaggle held the Jigsaw Unintended Bias in Toxicity Classification competition so that data scientists worldwide could work together to investigate ways to mitigate bias.

The code cell below loads some of the data from the competition. We will work with thousands of comments, where each comment is labeled as either "toxic" or "not toxic".

In [ ]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

# Get the same results each time
np.random.seed(0)

# Load the training data
data = pd.read_csv("data_4.csv")
comments = data["comment_text"]
target = (data["target"]>0.7).astype(int)

# Break into training and test sets
comments_train, comments_test, y_train, y_test = train_test_split(comments, target,

# Get vocabulary from training data
vectorizer = CountVectorizer()
vectorizer.fit(comments_train)

# Get word counts for training and test sets
X_train = vectorizer.transform(comments_train)
X_test = vectorizer.transform(comments_test)

# Preview the dataset
print("Data successfully loaded!\n")
print("Sample toxic comment:", comments_train.iloc[22])
print("Sample not-toxic comment:", comments_train.iloc[17])
```

```
Data successfully loaded!

Sample toxic comment: Too dumb to even answer.
Sample not-toxic comment: No they aren't.
```

Run the next code cell without changes to use the data to train a simple model. The output shows the accuracy of the model on some test data.

In [ ]:
```python
from sklearn.linear_model import LogisticRegression

# Train a model and evaluate performance on test dataset
classifier = LogisticRegression(max_iter=2000)
classifier.fit(X_train, y_train)
score = classifier.score(X_test, y_test)
print("Accuracy:", score)

# Function to classify any string
def classify_string(string, investigate=False):
    prediction = classifier.predict(vectorizer.transform([string]))[0]
    if prediction == 0:
        print("NOT TOXIC:", string)
    else:
        print("TOXIC:", string)
```

```
Accuracy: 0.9304755967877966
```

Roughly 93% of the comments in the test data are classified correctly!

Try out the model

You will use the next code cell to write your own comments and supply them to the model: does the model classify them as toxic?

1. Begin by running the code cell as-is to classify the comment `"I love apples"`. You should see that was classified as "NOT TOXIC".

2. Then, try out another comment: `"Apples are stupid"`. To do this, change only `"I love apples"` and leaving the rest of the code as-is. Make sure that your comment is enclosed in quotes, as below.

```python
my_comment = "Apples are stupid"
```

3. Try out several comments (not necessarily about apples!) to see how the model performs and if it performs as suspected.

Graded cell ►

In [ ]:
```python
# Comment to pass through the model
my_comment = "I hate this assignment, it is so stupid!"
other_comment = "I hate this assignment it so boring!"
comment_2 = "I am so sick of people posting comments about poltics on facebook, peo
```

```
comment_3 = "People are so dumb"
comment_4 = "Why do people have to be so dumb and post comments about politics on f
comment_5 = "Hey your code is great!"


# Do not change the code below
classify_string(my_comment)
classify_string(other_comment)
classify_string(comment_2)
classify_string(comment_3)
classify_string(comment_4)
classify_string(comment_5)
```

```
TOXIC: I hate this assignment, it is so stupid!
NOT TOXIC: I hate this assignment it so boring!
TOXIC: I am so sick of people posting comments about poltics on facebook, people who
do that are so dumb
TOXIC: People are so dumb
TOXIC: Why do people have to be so dumb and post comments about politics on faceboo
k, it is so annoying!
NOT TOXIC: Hey your code is great!
```

Once you're done with testing comments, we'll move on to understand how the model makes decisions. Run the next code cell without changes.

The model assigns each of roughly 58,000 words a coefficient, where higher coefficients denote words that the model thinks are more toxic. The code cell outputs the ten words that are considered most toxic, along with their coefficients.

In [ ]:
```
coefficients = pd.DataFrame({"word": sorted(list(vectorizer.vocabulary_.keys())), "
coefficients.sort_values(by=['coeff']).tail(10)
```

Out[ ]:

	word	coeff
20745	fools	6.279051
34211	moron	6.333047
16844	dumb	6.359822
12907	crap	6.490184
38317	pathetic	6.555196
25850	idiotic	7.005699
49802	stupidity	7.554397
25858	idiots	8.603087
25847	idiot	8.606546
49789	stupid	9.278841

Most toxic words

Take a look at the most toxic words from the code cell above. Are you surprised to see any of them?

Q. Are there any words that seem like they should not be in the list?

Graded cell ►

Response:I am honestly not suprised by any of the words other than the word 'Crap' becuse to me it seems like the model is just soley identifying verbs. I think that there needs to be some kind of context that should be applied to the words. I think that the model did a pretty good job to be fair.

A closer investigation

We'll take a closer look at how the model classifies comments.

1. Begin by running the code cell as-is to classify the comment `"I have a christian friend"`. You should see that was classified as "NOT TOXIC". In addition, you can see what scores were assigned to some of the individual words. Note that all words in the comment likely won't appear.
2. Next, try out another comment: `"I have a muslim friend"`. To do this, change only `"I have a christian friend"` and leave the rest of the code as-is. Make sure that your comment is enclosed in quotes, as below.

```
new_comment = "I have a muslim friend"
```

3. Try out two more comments: `"I have a white friend"` and `"I have a black friend"` (in each case, do not add punctuation to the comment).
4. Feel free to try out more comments to see how the model classifies them.

Graded cell ►

In [ ]:
```
# Set the value of new_comment
new_comment = "I have a black friend"

# Do not change the code below
classify_string(new_comment)
coefficients[coefficients.word.isin(new_comment.split())]
```

```
TOXIC: I have a black friend
```

Out[ ]:

	word	coeff
6893	black	2.104468
21256	friend	-0.130889
24049	have	-0.072310

Identify bias

In the code cell above,

- How did the model classify `"I have a christian friend"` and `"I have a muslim friend"`?
- How did it classify `"I have a white friend"` and `"I have a black friend"`?

Q. Do you see any signs of potential bias in the model?

Graded cell ►

Response: Yes I do see a very clear bias in the model. It showed that the words, 'Muslim' and 'Black' had a coeff, which means the model thinks negatively of those words.

Test your understanding

We'll step away from the Jigsaw competition data and consider a similar (but hypothetical!) scenario where you're working with a dataset of online comments to train a model to classify comments as toxic.

You notice that comments that refer to Islam are more likely to be toxic than comments that refer to other religions, because the online community is islamophobic.

Q. What type of bias can this introduce to your model?

Graded cell ►

Response: This kind of bias can be considered as representation bias, and will negatively effect the model and will give the model a false sense of accuracy because the data set is not cleaned enough to be deployed.

Test your understanding, part 2

We'll continue with the same hypothetical scenario, where you're trying to train a model to classify online comments as toxic.

You take any comments that are not already in English and translate them to English with a separate tool. Then, you treat all posts as if they were originally expressed in English.

Q. What type of bias will your model suffer from?

Graded cell ►

Response: There will be a translation bias in the model, and the model is limited because of the translation tool that is used. Sometimes those models are not very accurate, and there could be lots of issues when translating.

Test your understanding, part 3

We'll continue with the same hypothetical scenario, where you're trying to train a model to classify online comments as toxic.

The dataset you're using to train the model contains comments primarily from users based in the United Kingdom.

After training a model, you evaluate its performance with another dataset of comments, also primarily from users based in the United Kingdom -- and it gets great performance! You deploy it for a company based in Australia, and it does not perform well, because of differences between British and Australian English.

Q. What types of bias does the model suffer from?

Graded cell ►

Response:The model will actually be experincing regional bias or geography bias, because the UK might have diffrent language rules, than Australia. The model will have a hard time workiing properly, all due to the fact that people talk diffrently in other parts of the world.

Learn more

To continue learning about bias, check out the Jigsaw Unintended Bias in Toxicity Classification competition that was introduced in this exercise.

- Kaggler Dieter has written a helpful two-part series that teaches you how to preprocess the data and train a neural network to make a competition submission. Get started here.
- Many Kagglers have written helpful notebooks that you can use to get started. Check them out on the competition page.

Another Kaggle competition that you can use to learn about bias is the Inclusive Images Challenge, which you can read more about in this blog post. The competition focuses on **evaluation bias** in computer vision.

Keep going

How can you quantify bias in machine learning applications? Continue to **learn how to measure fairness**.