1. When using the toxicity library, a statement will be labelled with 2 probabilities. What are they? **1 / 1 point**

- ○ The first is the probability value for whether or not the phrase is an insult, and the second is the probability for whether or not it is not
- ○ The first is the probability value for whether or not the phrase is an insult, and the second is the threshold
- ○ The first is the probability value for whether or not the phrase is not an insult, and the second is the threshold
- ◉ The first is the probability value for whether or not the phrase is not an insult, and the second is the probability for whether or not it is

✓ **Correct**

2. If toxicity returns a probabilities list with values of [0.8, 0.2], what does that mean? **1 / 1 point**

- ○ The phrase does not contain an insult
- ○ The phrase contains an insult
- ○ There's an error
- ◉ We don't know. The answer depends on something else

✓ **Correct**

3. How do you determine what type of toxicity is contained in a result from toxicity? **1 / 1 point**

- ○ When you call the API you specify what type of toxicity you are looking for with a parameter (i.e. 'threat')
- ◉ It returns an array of answers, each one corresponding to a different type of toxicity
- ○ There's no way to determine type of toxicity, either a sentence is toxic or it isn't
- ○ When you call the API you send it a list of specific toxicity types you want it to look for (i.e. (['threat', 'obscene'])

✓ **Correct**

4. When using mobilenet in js to classify an image, it can recognize up to 1000 types. How many predictions does it return by default? **1 / 1 point**

- ○ All that are above a threshold, set by the threshold parameter
- ○ 1000
- ◉ 3
- ○ All non-zero predictions

✓ **Correct**

5. When converting Python-trained models to JSON to use in tensorflow.js, what is the package that you need to 'pip install' (assuming you already have installed tensorflow)

- ○ None, it's built into TensorFlow
- ○ tensorflow-js
- ◉ Tensorflowjs
- ○ tensorflow-javascript

✓ **Correct**

6. How do you convert a Python-trained model to JSON?

- ◉ Save it as a TensorFlow Saved Model, then use the tensorflowjs_convertor script in Python
- ○ Simply save it as JSON
- ○ Save it as a TensorFlow Saved Model, then import that as a JSON object
- ○ Save it as a TensorFlow Saved Model, then use the tensorflowjs_convertor script in JavaScript

✓ **Correct**

7. If you have a model that you've converted to JSON how do you load it into JavaScript?

- (●)
```
const model = await tf.loadLayersModel(MODEL_URL)
```
- ( )
```
const model = tf.loadLayersModel(MODEL_URL)
```
- ( )
```
const model = tf.loadSavedModel(MODEL_URL)
```
- ( )
```
const model = await tf.loadSavedModel(MODEL_URL)
```

8. When you convert a Python-based model to JSON, how many files will you get?

- ( ) One, the model file itself
- (●) At least two: the model file, and a sharded collection of binary weight files that can have one or more files
- ( ) Two, the model file and a metadata file
- ( ) Two, the model file and a snapshot of binary weights

✓ **Correct**