What you will do

Now you are ready! We are going do three tasks in this assignment. There are several results you need to gather along the way to enter into the quiz after this reading.

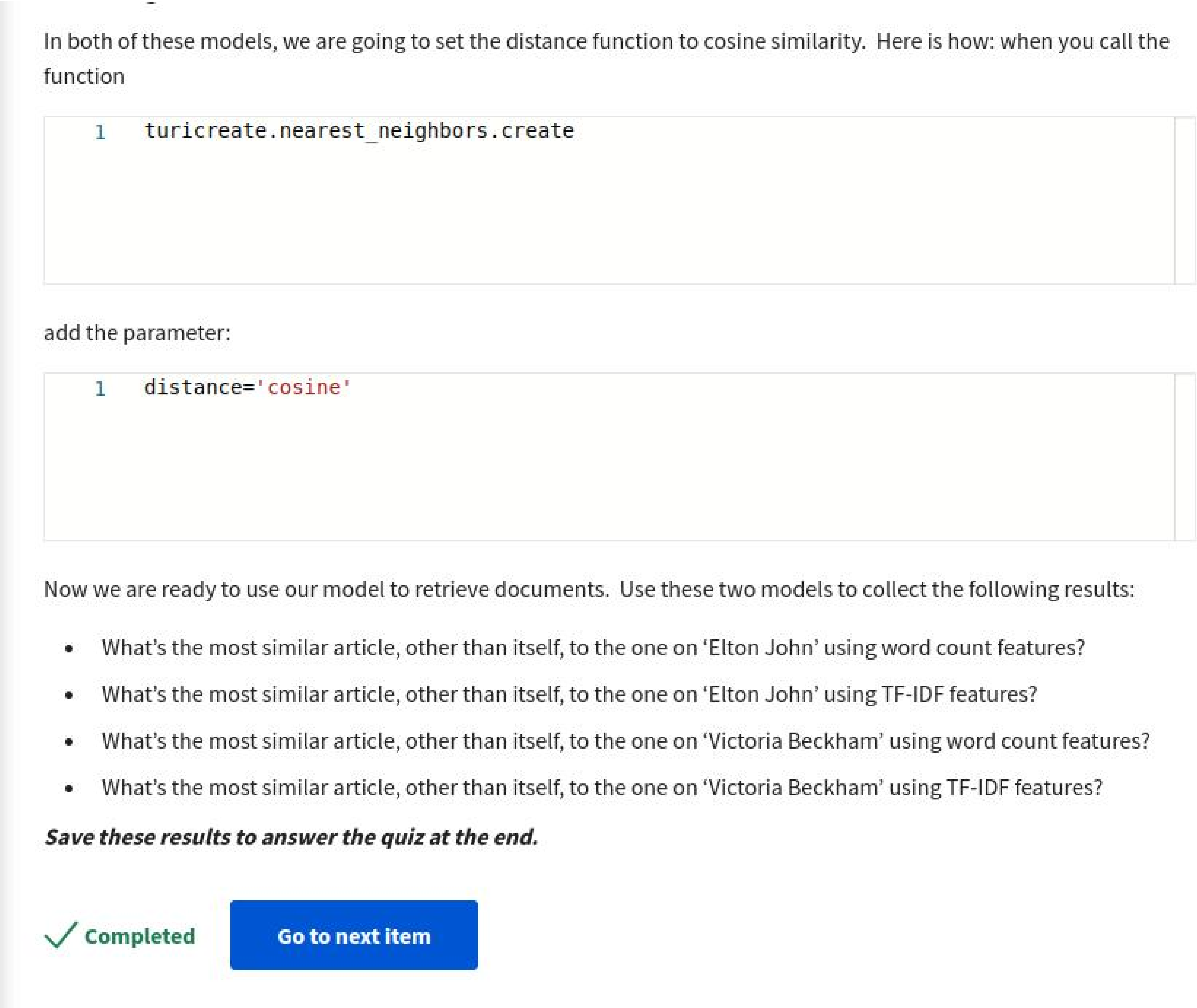
1. **Compare top words according to word counts to TF-IDF:** In the notebook we covered in the module, we explored two document representations: word counts and TF-IDF. Now, take a particular famous person, 'Elton John'. What are the 3 words in his articles with highest word counts? What are the 3 words in his articles with highest TF-IDF? These results illustrate why TF-IDF is useful for finding important words. ***Save these results to answer the quiz at the end.***
2. **Measuring distance:** Elton John is a famous singer; let's compute the distance between his article and those of two other famous singers. In this assignment, you will use the *cosine distance*, which one measure of similarity between vectors, similar to the one discussed in the lectures. You can compute this distance using the *turicreate.distances.cosine* function. What's the cosine distance between the articles on 'Elton John' and 'Victoria Beckham'? What's the cosine distance between the articles on 'Elton John' and Paul McCartney'? Which one of the two is closest to Elton John? Does this result make sense to you? ***Save these results to answer the quiz at the end.***
3. **Building nearest neighbors models with different input features and setting the distance metric:** In the sample notebook, we built a nearest neighbors model for retrieving articles using TF-IDF as features and using the default setting in the construction of the nearest neighbors model. Now, you will build two nearest neighbors models:

- Using word counts as features
- Using TF-IDF as features

In both of these models, we are going to set the distance function to cosine similarity. Here is how: when you call the function

```
turicreate.nearest_neighbors.create
```

add the parameter:



In both of these models, we are going to set the distance function to cosine similarity. Here is how: when you call the function

```
turicreate.nearest_neighbors.create
```

add the parameter:

```
distance='cosine'
```

Now we are ready to use our model to retrieve documents. Use these two models to collect the following results:

- What's the most similar article, other than itself, to the one on 'Elton John' using word count features?
- What's the most similar article, other than itself, to the one on 'Elton John' using TF-IDF features?
- What's the most similar article, other than itself, to the one on 'Victoria Beckham' using word count features?
- What's the most similar article, other than itself, to the one on 'Victoria Beckham' using TF-IDF features?

Save these results to answer the quiz at the end.

✓ Congratulations! You passed!

Grade received 100%

Latest Submission Grade 100%

To pass 80% or higher

Go to next item

1. **Top word count words for Elton John** — 1 / 1 point

   ✓ Correct

2. **Top TF-IDF words for Elton John** — 1 / 1 point

   ✓ Correct

3. **The cosine distance between 'Elton John's and 'Victoria Beckham's articles (represented with TF-IDF) falls within which range?** — 1 / 1 point

   ✓ Correct

4. **The cosine distance between 'Elton John's and 'Paul McCartney's articles (represented with TF-IDF) falls within which range?** — 1 / 1 point

   ✓ Correct

5. **Who is closer to 'Elton John', 'Victoria Beckham' or 'Paul McCartney'?**

1 / 1 point

✓ **Correct**

6. **Who is the nearest cosine-distance neighbor to 'Elton John' using raw word counts?**

1 / 1 point

✓ **Correct**

7. **Who is the nearest cosine-distance neighbor to 'Elton John' using TF-IDF?**

1 / 1 point

✓ **Correct**

8. **Who is the nearest cosine-distance neighbor to 'Victoria Beckham' using raw word counts?**

1 / 1 point

✓ **Correct**

9. **Who is the nearest cosine-distance neighbor to 'Victoria Beckham' using TF-IDF?**

1 / 1 point

✓ **Correct**