Congratulations! You passed!

Grade received 80% To pass 80% or higher

Go to next item

1.	If you want to merge two splits 'train' and 'test' together using Splits API, how would you be able to do so?	1/1 point
	tfds.load('mnist', split = 'train + test')	
	tfds.load('mnist', merge = 'train+test')	
	<pre>tfds.load('mnist',split = np.concat('train+test'))</pre>	
	tfds.load('mnist', split = pd.concat('train', 'test'))	
	Correct!	
	Passing both train and test in the string is the proper way to get both the splits.	
2.	The MNISTv3 dataset supports the Splits API. The train split has 70000 records in it. If you just want to create a subsplit of the first 7000 records and want to use the python slicing notation instead of Splits API, what would be the answer?	1/1 point
	tfds.load('mnist:3.*.*', split='train[:7000]')	
	tfds.load('mnist:3.*.*', split='train[7000:]')	
	Read the entire train split, create a new dataset, iterate over the first 7000 of the 70000, and copy the records one-by-one to the new dataset.	
	tfds.load('mnist:3.*.*', subsplit='train[:7000]')	
	Correct!	
	train[:7000] technically takes records from 0 to 6999 index value.	
3.	If you want a subsplit of the first 10% of the MNISTv3 training records, what would the code look like using the Splits API?	1/1 point
	tfds.load('mnist:3.*.*', subsplit='train[:10%]')	
	tfds.load('mnist:3.*.*', subsplit='train[10%:]')	
	tfds.load('mnist:3.*.*', split='train[:10%]')	
	tfds.load('mnist:3.*.*', split='train[10%:]')	
	Correct!	
	'train[:10%]'in string format represents that we want the first 10% of the records from the train split.	
4.	How many validation splits will this code generate?	1/1 point
	val_ds = tfds.load('mnist:3.*.*', split = ['train[{}%:{}%]'.format(int(k/4),int((k+40)/4)) for k in range(0,400,40)])	
	O Will throw an Error	
	O 5	
	10	
	O 40	
	✓ Correct Correct!	
	As k is incremented by 40, you get the values like (0,40), (40,80) until the last one, (360:400).	
	Dividing each value by 4 as you have (k/4,(k+40)/4), it will get converted to [0%:10%],[10%:20%][90%:100%] which is 10 splits. Note that the indices should be integers so the <i>int()</i> function was used inside the list comprehension to do that conversion.	

Table ○ Inserred Sea fining is based on the dange size of the sales. If the dataset itself is in GNs, you can expect the solication and test servado to be bigger and probably receifing sharings. 6. Which of the following could be used to investigate now the TTRecords (like sold like?) © Inserred "your If record file" raw file "It the salat aced boad (literature) for raw record in raw file state) (i) print (epitras record) □ Illenared "your If record file" raw file "It be each raw do different ratio" raw, record in raw file state) (i) print (epitras record) □ Illenared "your If record file" raw file "It boads (literature) raw, record in raw, file state) (i) print (epitras record) □ Illenared "your If record file" raw file "It boads (literature) raw, record in raw, file table); print (epitras record) □ Convect □ Convect □ It data TTRecord Datase is used to rear your raw TTRecord Bias and convert it to an object. □ Late(1) gives your "row in run the raw, file TTRecord Datase object. □ Late(1) gives your "row in run the raw, file TTRecord Datase object. □ The record record record of the low raw TTRecord Ties look like other your properly read and print them retill "TTRecord Dataset methods. □ Convect □ Convect □ Convect (a shape(1), disyperating, numpy via "not (nin x1) in x86 feature x2 \times x2 \tim	0/1 point
Sharing is based on the stanger size of the splies. If the dataset inset is in Gills, you can expect the solidation and rest sets also to be bigger and probably needing that ding. 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Which of the following could be used to investigate how the TPRecordSides (Restook) [ie?] 6. Earname "your, if, record, file "your, file elitions (Restook) [ie] are greated in any, file the file (I); printiply (I); printi	
filesame = "your_tt_record_file" aw_file = tt dataTFRecord batasetfilename(for raw_record in raw_file taset1); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record)) filesame = "your_tt_record_file" aw_file = tt floads filename(for raw_record in raw_file taket(t); print(repr):raw_record) filesame = "your_tt_record_file" aw	
filename = "your_tt_record_file"row_file = tt.deas.TFRecord.Datasetfilename/for row_record in row_file_taxet(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deas.deasetd.odd?ilename/for row_record in row_file_taxet(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.filename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfilename/for row_record in row_file_taket(1): print(repr):row_record)) filename = "your_tt_record_file"row_file = tt.deacd.parasetfile_file_file_file_file_file_file_file_	
filename = "your_st_record_file"raw_file = tt.lecas_dataset.load/litenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_files_take(1): print(repriraw_record) filename = tt.lecas_file_take(1):	1/1 point
filename = "your_st_record_file"raw_file = tt.lecas_dataset.load/litenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_file"raw_file = tt.lecas_filenameifor raw_record in raw_file_take(1): print(repriraw_record)) filename = "your_st_record_files_take(1): print(repriraw_record) filename = tt.lecas_file_take(1):	
filename = "your_tf_record_file"raw_file = tf.loads(filename)for raw_record in raw_file_take(1); print(repr(raw_record)) filename = "your_tf_record_file"raw_file = tf.RecordDataset/filename/for raw_record in raw_file_take(1); print(repr(raw_record)) Correct	
filename = "your_st_record_file"raw_file = st.RecordDataset(filename)for raw_record in raw_file.take(1): print(repr(raw_record)) Correct Correct It dataTFRecordDataset is used to read your raw TFRecord files and convert it to an object. take(1) gives you a record from the raw_file TFRecord Dataset object. repr(1) function returns a printable representation of the given strecord object. 7. Bollow is an example of how raw TFRecord files look like when you properly read and print them with TFRecordDataset methods.	
Correct Correct! tt.data.TFRecordDataset is used to read your raw TFRecord files and convert it to an object. take(1) gives you 1 record from the raw_file TFRecordDataset object. repri) function returns a printable representation of the given tfrecord object. 7. Below is an example of how raw TFRecord files look like when you properly read and print them with TFRecordDataset methods. <tf.tensor: \n0\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n<="" dtype="string," numpy="b" shape="()," td=""><td></td></tf.tensor:>	
t. take[1] gives you 1 record from the raw_file TFRecord files and convertit to an object. take[1] gives you 1 record from the raw_file TFRecord baseet object. repri) function returns a printable representation of the given threcord object. 7. Below is an example of how raw TFRecord files look like when you properly read and print them with TFRecord baseet methods. </td <td></td>	
t. It. data. TRecord Dataset is used to read your raw TFRecord files and convert it to an object. take(1) gives you 1 record from the raw_file TFRecord Dataset object. repri) function returns a printable representation of the given tfrecord object. 7. Below is an example of how raw TFRecord files look like when you properly read and print them with TFRecordDataset methods. <pre> </pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> <pre></pre></pre></pre></pre></pre></pre>	
repr() function returns a printable representation of the given threcord object. 7. Below is an example of how raw TFRecord files look like when you properly read and print them with TFRecordDataset methods. **Tf.Tensor: shape=(), dtype=string, numpy=b'\nQ\n\x13\n\x88feature2\x12\x87\n\x85\n\x85\n\x83\n\x14 tf.Tensor: shape=(), dtype=string, numpy=b'\nS\n\x15\n\x85\n\	
To parse them into proper format, which of the following options need to be implemented? Apply the parsing function to each item in the dataset using thetf.data.Dataset.mapmethod. Creating a parsing function using tf.io.parse_single_example() is used to parse examples one by one as the raw TFRecordDataset files contain serialized tit.ria. Stap 1. You need to define your feature descriptions to build their shape and type signature Creating a feature description dictionary Creating a feature description dictionary Creating a feature description dictionary Creating a feature description to build their shape and type signature Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata.This is necessary here because datasets use graph-execution, and need description to build their shape and type signature Leg and type signature Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata.This is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
<pre></pre> <pre><pre></pre> <pre></pre> <pre></pre> <pre><pre></pre> <pre></pre> <pre><pre></pre> <pre><pre><pre></pre> <pre><pre><pre></pre> <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>	
<pre></pre> <pre></pre> <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	
<pre></pre> <pre></pre> <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	
 Apply the parsing function to each item in the dataset using thetf.data.Dataset.mapmethod. ✓ Creating a parsing function using tf.io.parse_single_example() ✓ Correct Correct Correct This is Step 2. tf.io.parse_single_example() is used to parse examples one by one as the raw TFRecordDataset files contain serialized tf.train.Example objects. You can also use tf.parse_example to parse the whole batch at once. ✓ Creating a feature description dictionary ✓ Correct Correct Correct Correct This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This is necessary here because datasets use graph-execution, and need description to build their shape and type signature 	0.6 / 1 point
 ✓ Correct Correct This is Step 2. **tf.io.parse_single_example() is used to parse examples one by one as the raw TFRecordDataset files contain serialized tf.train.Example objects. You can also use **tf.parse_example** to parse the whole batch at once. ✓ Correct Correct Correct This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This is necessary here because datasets use graph-execution, and need description to build their shape and type signature 	
 ✓ Correct Correct This is Step 2. tf.io.parse_single_example() is used to parse examples one by one as the raw TFRecordDataset files contain serialized tf.train.Example objects. You can also use tf.parse_example to parse the whole batch at once. ✓ Creating a feature description dictionary ✓ Correct Correct Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This is necessary here because datasets use graph-execution, and need description to build their shape and type signature 	
Correct! This is Step 2. **tf.io.parse_single_example()** is used to parse examples one by one as the raw TFRecordDataset files contain serialized tf.train.Example objects. You can also use **tf.parse_example** to parse the whole batch at once. **Creating a feature description dictionary** **Correct** Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata.This is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
tf.train.Example objects. You can also use tf.parse_example to parse the whole batch at once. Creating a feature description dictionary Correct Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This Is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
Correct Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This Is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This Is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
Correct! This is Step 1. You need to define your feature descriptions properly based on the dataset and its metadata. This is necessary here because datasets use graph-execution, and need description to build their shape and type signature	
use graph-execution, and need description to build their shape and type signature	
☐ Creating a parsing function using tfds.load()	
Apply the parsing function to each item in the dataset using thekeras.dataset.map method	

Page 3

Week 2 | Coursera

https://www.coursera.org/learn/data-pipelines-tensorflow/quiz/84GiW/week-2/view-attempt



X This should not be selected

map method is not present in the keras.dataset class.