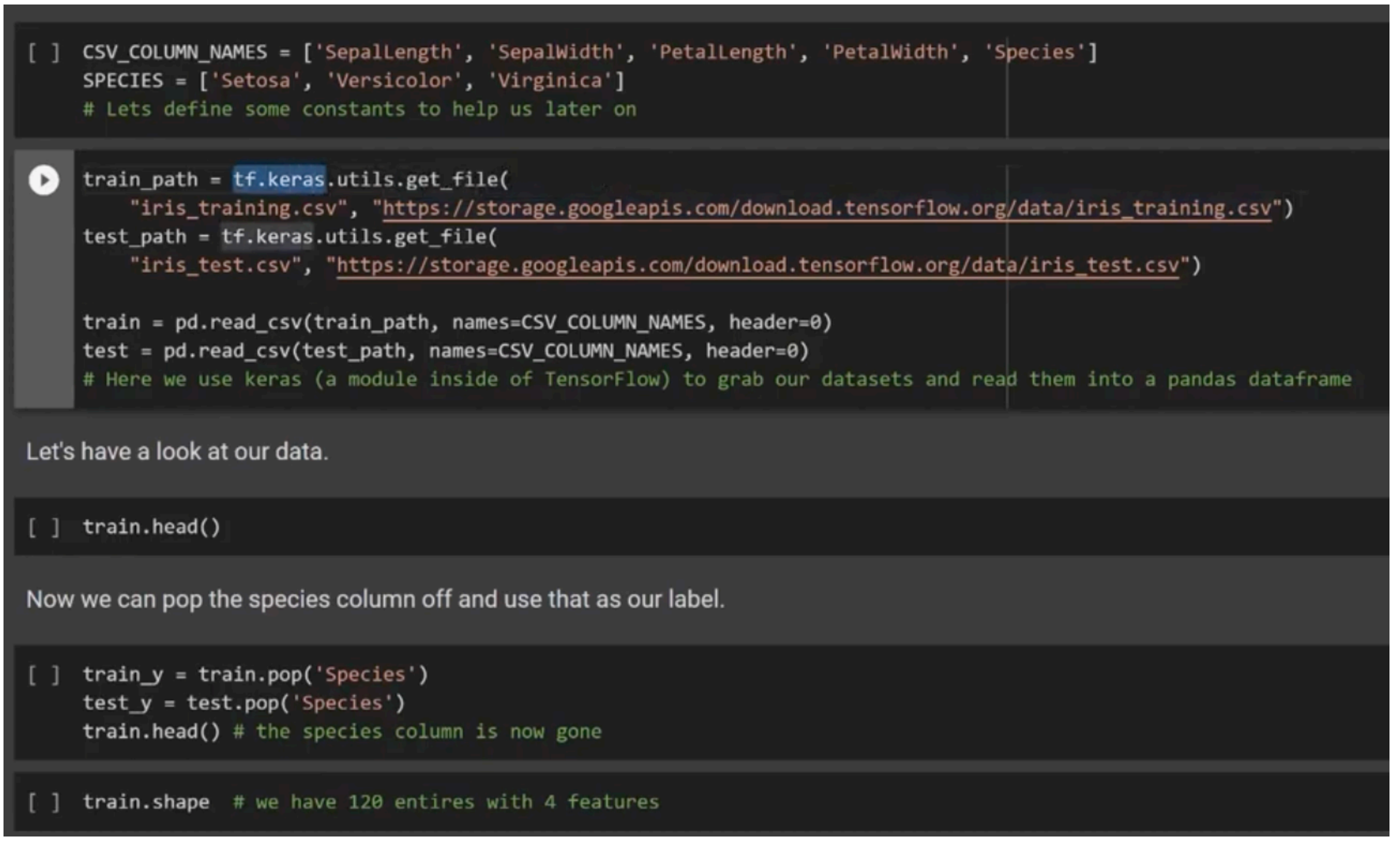
- ***Classification***
  - -> separating different datapoints into classes
  - -> linear regression outputs a value and classification puts datapoints into classes
  - -> there are a lot of different algorithms which do this
- ***Example for the Iris dataset***
  - ***Importing the data***



```
[ ] CSV_COLUMN_NAMES = ['SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth', 'Species']
    SPECIES = ['Setosa', 'Versicolor', 'Virginica']
    # Lets define some constants to help us later on
```

```
train_path = tf.keras.utils.get_file(
    "iris_training.csv", "https://storage.googleapis.com/download.tensorflow.org/data/iris_training.csv")
test_path = tf.keras.utils.get_file(
    "iris_test.csv", "https://storage.googleapis.com/download.tensorflow.org/data/iris_test.csv")

train = pd.read_csv(train_path, names=CSV_COLUMN_NAMES, header=0)
test = pd.read_csv(test_path, names=CSV_COLUMN_NAMES, header=0)
# Here we use keras (a module inside of TensorFlow) to grab our datasets and read them into a pandas dataframe
```

Let's have a look at our data.

```
[ ] train.head()
```

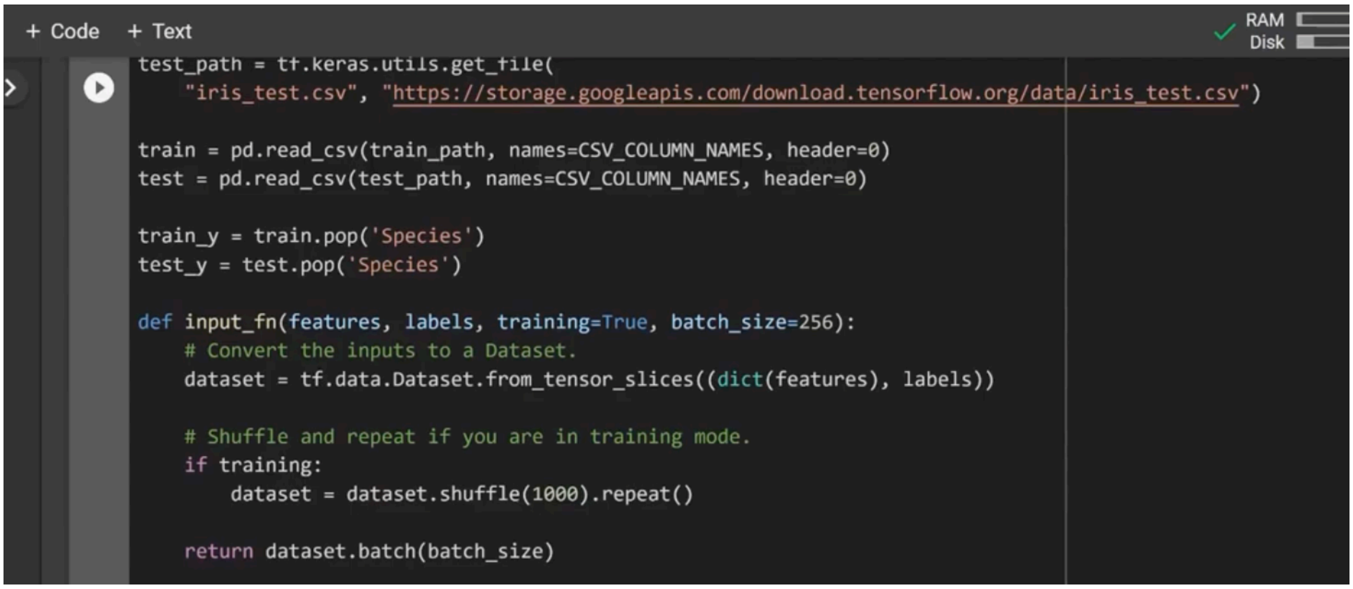
Now we can pop the species column off and use that as our label.

```
[ ] train_y = train.pop('Species')
    test_y = test.pop('Species')
    train.head() # the species column is now gone
```

```
[ ] train.shape  # we have 120 entires with 4 features
```

- -> he's imported tensor flow and the modules
- -> then imported the flower dataset
  - the petal width, length etc
- -> the csv column names and species, then datasets
- -> it's loading the csv file with the data
  - -> row 0 is the header
- -> he is using keras to import the data into the project

  - ***Exploring the data***



```
test_path = tf.keras.utils.get_file(
    "iris_test.csv", "https://storage.googleapis.com/download.tensorflow.org/data/iris_test.csv")

train = pd.read_csv(train_path, names=CSV_COLUMN_NAMES, header=0)
test = pd.read_csv(test_path, names=CSV_COLUMN_NAMES, header=0)

train_y = train.pop('Species')
test_y = test.pop('Species')

def input_fn(features, labels, training=True, batch_size=256):
    # Convert the inputs to a Dataset.
    dataset = tf.data.Dataset.from_tensor_slices((dict(features), labels))

    # Shuffle and repeat if you are in training mode.
    if training:
        dataset = dataset.shuffle(1000).repeat()

    return dataset.batch(batch_size)
```

- -> he's printed out the header
- -> it's a dataset with values for the different features of the plant

- -> there are three different flower species

- ***Then he pops off the plant species***
  - -> what we want to predict
  - -> he's looking at the shape of the training data
  - -> train.shape <- this is for the data which trains the model
  - -> he's shuffled the dataset when training it
- ***He then makes an input function for the data***
  - -> there are no epochs and the batch sizes are different
  - -> in other words when it comes to training the model - the data isn't divided into batches to train it on
  - -> dataset = dataset.shuffle
  - -> he defined an input function with the different labels for the model
- ***Then for the feature columns***
  - -> i.e turning the names of plant species into 1s and 0s rather than words
  - -> coding this into the dataset
  - -> then he's iterating through the entire thing
  - -> classification is the process of separating data points into different classes
  - -> the plant species in the dataset are already encoded as 1, 2 and 3 (rather than the names of the plants)
    - he's takes these and appended them t the different columns he's iterating through