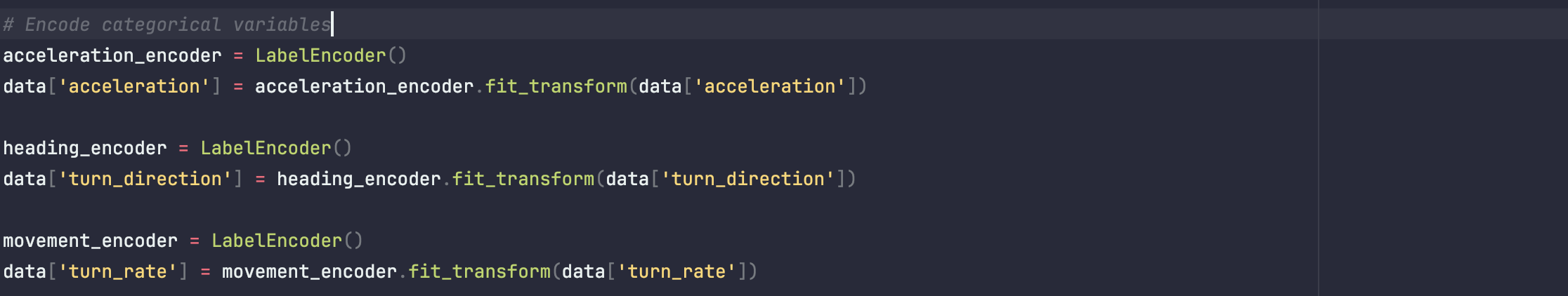
## Aircraft Trajectory Prediction using LSTM: Code Documentation

The code begins by loading the data from a CSV file using pandas:



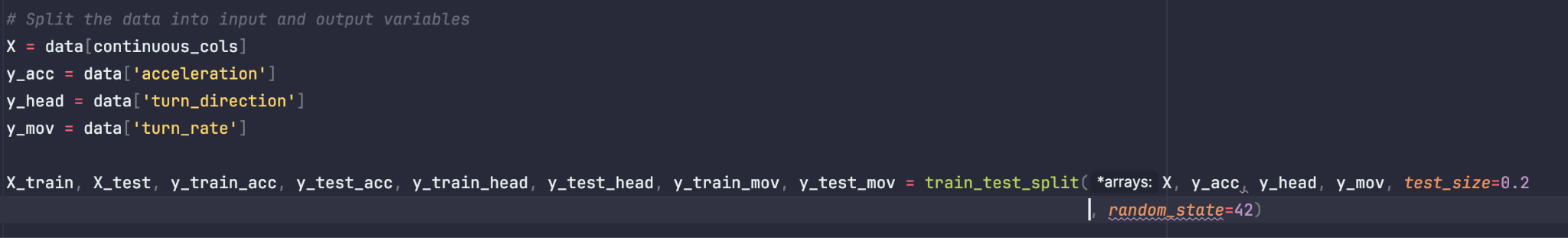
Then, the categorical variables (acceleration, heading, and movement) are encoded using the LabelEncoder from scikit-learn:



The continuous variables (latitude, longitude, position, course, speed) are normalized using the MinMaxScaler from scikit-learn:



Next, the data is split into input features (X) and output labels (y\_acc, y\_head, y\_mov) for acceleration, heading, and movement, respectively, using train\_test\_split from scikit-learn:

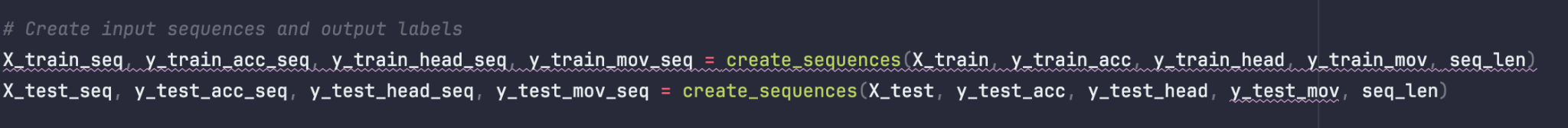


The create\_sequences function is defined to create input sequences and corresponding output labels for the LSTM model:



This function creates overlapping sequences of length seq\_len from the input data, and the corresponding output labels are taken from the next time step.

The input sequences and output labels are converted to PyTorch tensors:

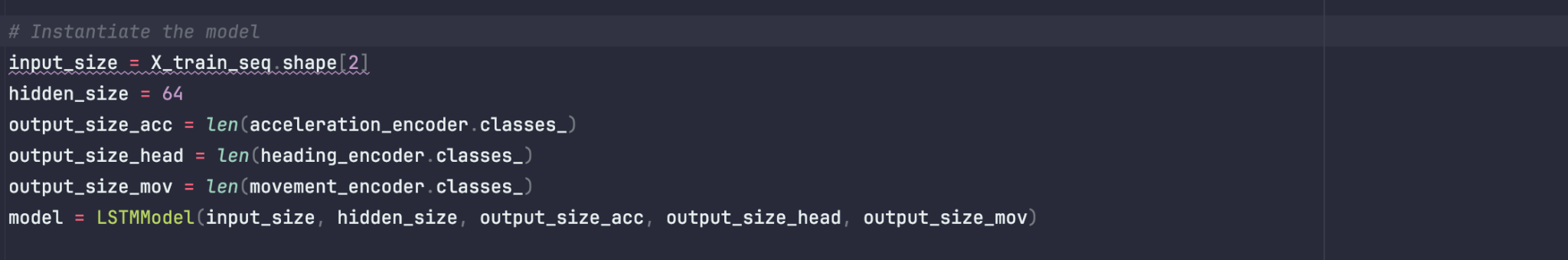


The LSTM model is defined as a PyTorch module called LSTMModel:



The model takes an input sequence of shape (batch\_size, seq\_len, input\_size), passes it through an LSTM layer, and then uses three separate fully connected layers to produce the predictions for acceleration, heading, and movement, respectively.

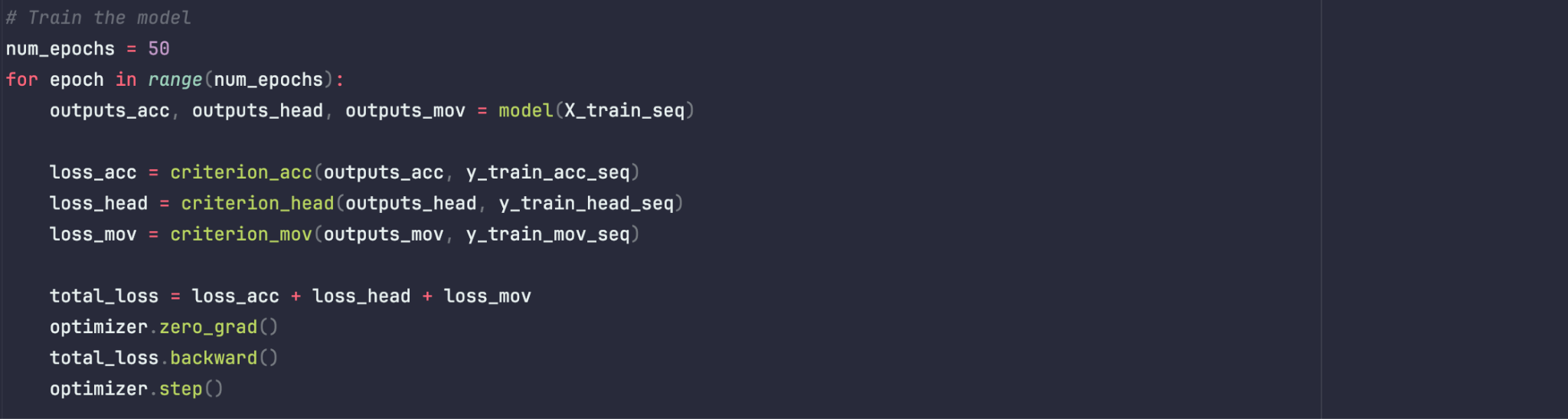
The model is instantiated with appropriate input size, hidden size, and output sizes:



Cross-entropy loss functions and the Adam optimizer are defined for training:



The model is trained for a specified number of epochs (num\_epochs):



Finally, the model is evaluated on the test data, and the accuracy for acceleration, heading, and movement predictions is calculated and printed:



This solution approach leverages the strengths of LSTM models in capturing long-term dependencies and modeling sequential data, making it suitable for predicting the future states of aircraft based on their past trajectories.

Output:

it's crucial to highlight an important concern regarding the accuracy of the LSTM model we implemented for the aircraft trajectory prediction problem. As mentioned in your previous email, the values for acceleration, turn\_direction (which we considered as heading), and turn\_rate (which we considered as movement) are all zero for all rows in the dataset. This means that the model is essentially trying to learn from constant values, which may lead to inaccurate predictions and artificially high or low accuracies reported by the model, in this case we are getting 100% accuracy

