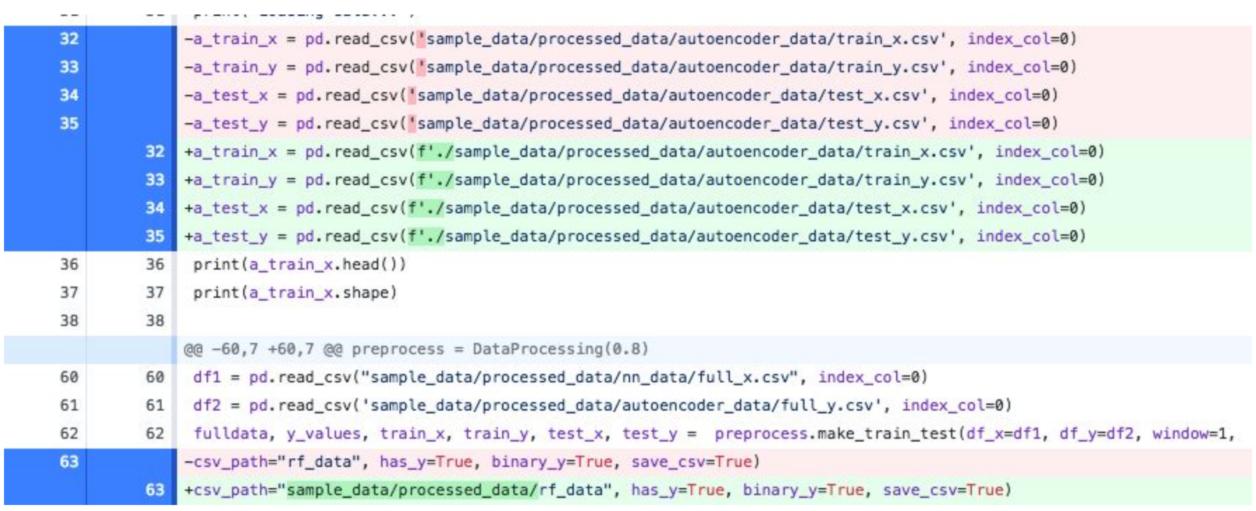
Andy Liu

Autoencoder (AE) is an unsupervised data representation method, similar to methods such as Principal Component Analysis. It can be used for feature selection and dimensionality reduction; however, it has the advantage of being nonlinear, and therefore more easily flexible when considering different data. Thus, less financial expertise is needed by the creator of the model. Autoencoders are, generally speaking, neural networks with "encoder" and "decoder" neuron components. These neural networks will learn a "reduction" side, where dimensionality is reduced, and a "reconstruction" side, where the original data is recovered from the reduced data. The parameters of the model are constructed by minimizing reconstruction error, or the distance from the reconstructed data to the original data.

The AIAlpha project used an autoencoder combined with an LSTM to predict stock prices and also ran the LSTM without autoencoder data in order to demonstrate the impact of the autoencoder. The project's creator claimed a log loss of **2.56** with the autoencoder and **3.62** without the autoencoder, compared to a base log loss of 2.94. When rerunning the code (with minor edits to file paths so that it could still run), log loss values of **4.41** (with autoencoder) and **5.02** (without autoencoder) were found. However, this was done with a very limited public dataset, compared to a far larger dataset used by the AIAlpha project creator, due to Github size restrictions, so judgment should be withheld until we can train a similar model on our own financial data.

Specific changes made to code:



```
-a_train_x = pd.read_csv('sample_data/processed_data/autoencoder_data/train_x.csv', index_col=0)
-a_train_y = pd.read_csv('sample_data/processed_data/autoencoder_data/train_y.csv', index_col=0)
-a_test_x = pd.read_csv('sample_data/processed_data/autoencoder_data/test_x.csv', index_col=0)
-a_test_y = pd.read_csv('sample_data/processed_data/autoencoder_data/test_y.csv', index_col=0)
+a_train_x = pd.read_csv(f'./sample_data/processed_data/autoencoder_data/train_x.csv', index_col=0)
+a_train_y = pd.read_csv(f'./sample_data/processed_data/autoencoder_data/train_y.csv', index_col=0)
+a_test_x = pd.read_csv(f'./sample_data/processed_data/autoencoder_data/test_x.csv', index_col=0)
+a_test_y = pd.read_csv(f'./sample_data/processed_data/autoencoder_data/test_y.csv', index_col=0)
 print(a_train_x.head())
 print(a_train_x.shape)

@@ -60,7 +60,7 @@ preprocess = DataProcessing(0.8)
 df1 = pd.read_csv("sample_data/processed_data/nn_data/full_x.csv", index_col=0)
 df2 = pd.read_csv('sample_data/processed_data/autoencoder_data/full_y.csv', index_col=0)
 fulldata, y_values, train_x, train_y, test_x, test_y =  preprocess.make_train_test(df_x=df1, df_y=df2, window=1,
-csv_path="rf_data", has_y=True, binary_y=True, save_csv=True)
+csv_path="sample_data/processed_data/rf_data", has_y=True, binary_y=True, save_csv=True)
```