Normalize the data

Why?

Normalization in machine learning is a preprocessing technique used to scale the features of a dataset to a similar range. There are several reasons why normalization is important:

1. **Improving Convergence**: Many machine learning algorithms, particularly those based on gradient descent, converge faster when features are normalized. This is because normalization helps to avoid extremely large or small gradients which can slow down convergence or prevent it altogether.
2. **Ensuring Fair Comparison**: Normalization ensures that all features contribute equally to the computation of distances or similarities. Without normalization, features with larger scales may dominate the learning process, leading to biased models.
3. **Handling Different Scales**: Features in a dataset often have different units or scales. For example, one feature may range from 0 to 100 while another ranges from 0 to 1. Normalization brings these features to a common scale, preventing features with larger scales from overshadowing those with smaller scales.
4. **Regularization**: In some regularization techniques, such as L1 and L2 regularization, the scale of features can affect the regularization term's impact on the model. Normalizing features can ensure that regularization is applied uniformly across all features.
5. **Improving Interpretability**: Normalization can make the interpretation of coefficients or feature importance scores easier. When features are on a similar scale, it's easier to compare their coefficients or importance values directly.

There are various methods of normalization, such as min-max scaling, z-score normalization (standardization), and robust scaling, each suitable for different types of data and algorithms. The choice of normalization method depends on the specific characteristics of the dataset and the requirements of the machine learning algorithm being used.

Flattening

X\_train shape (60000, 784)

y\_train shape (60000, 10)

we already had a 1d array with 784 rows. This represents the input layer.

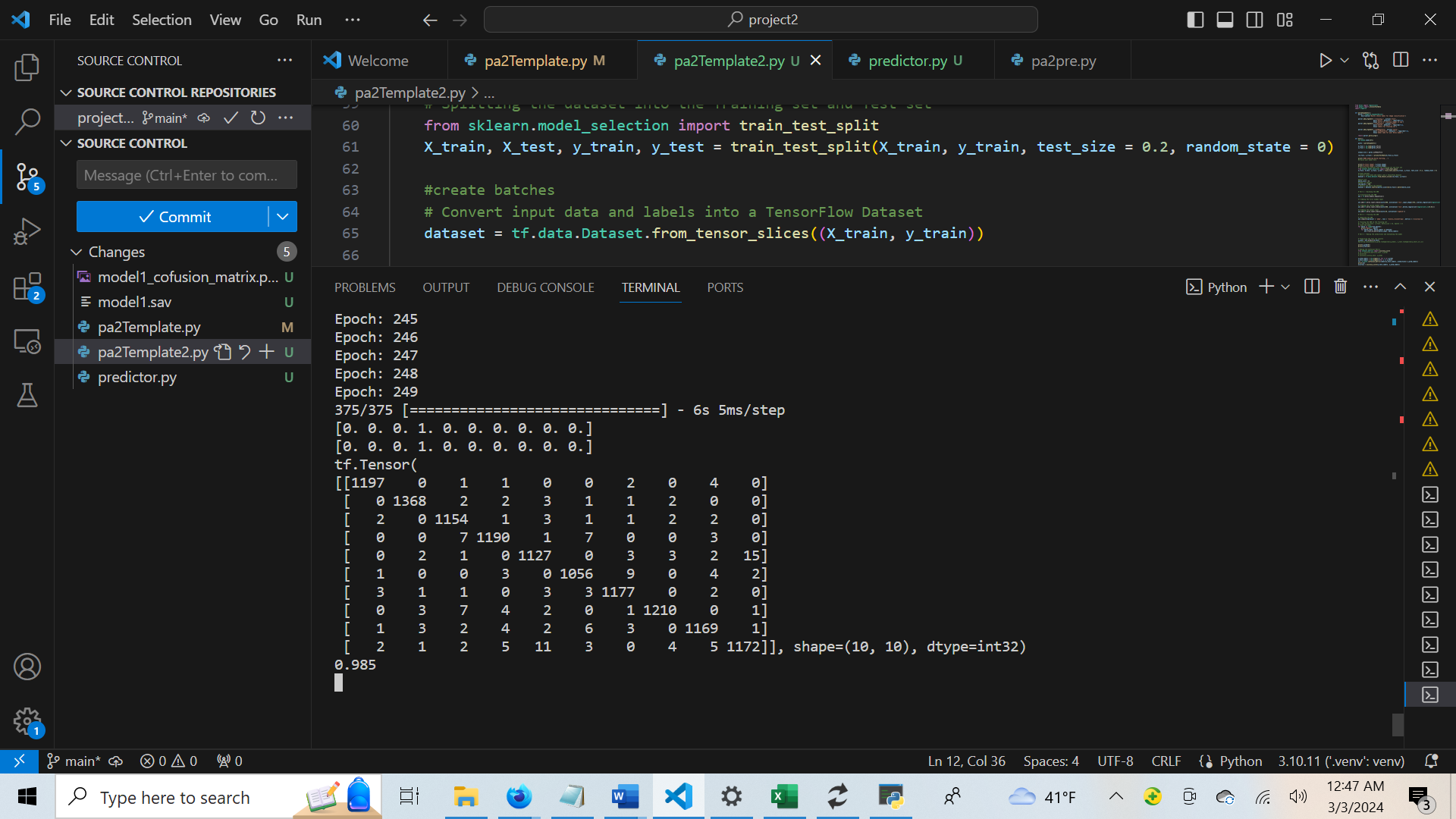
**Why two layers?** <https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/>

**Dropout**

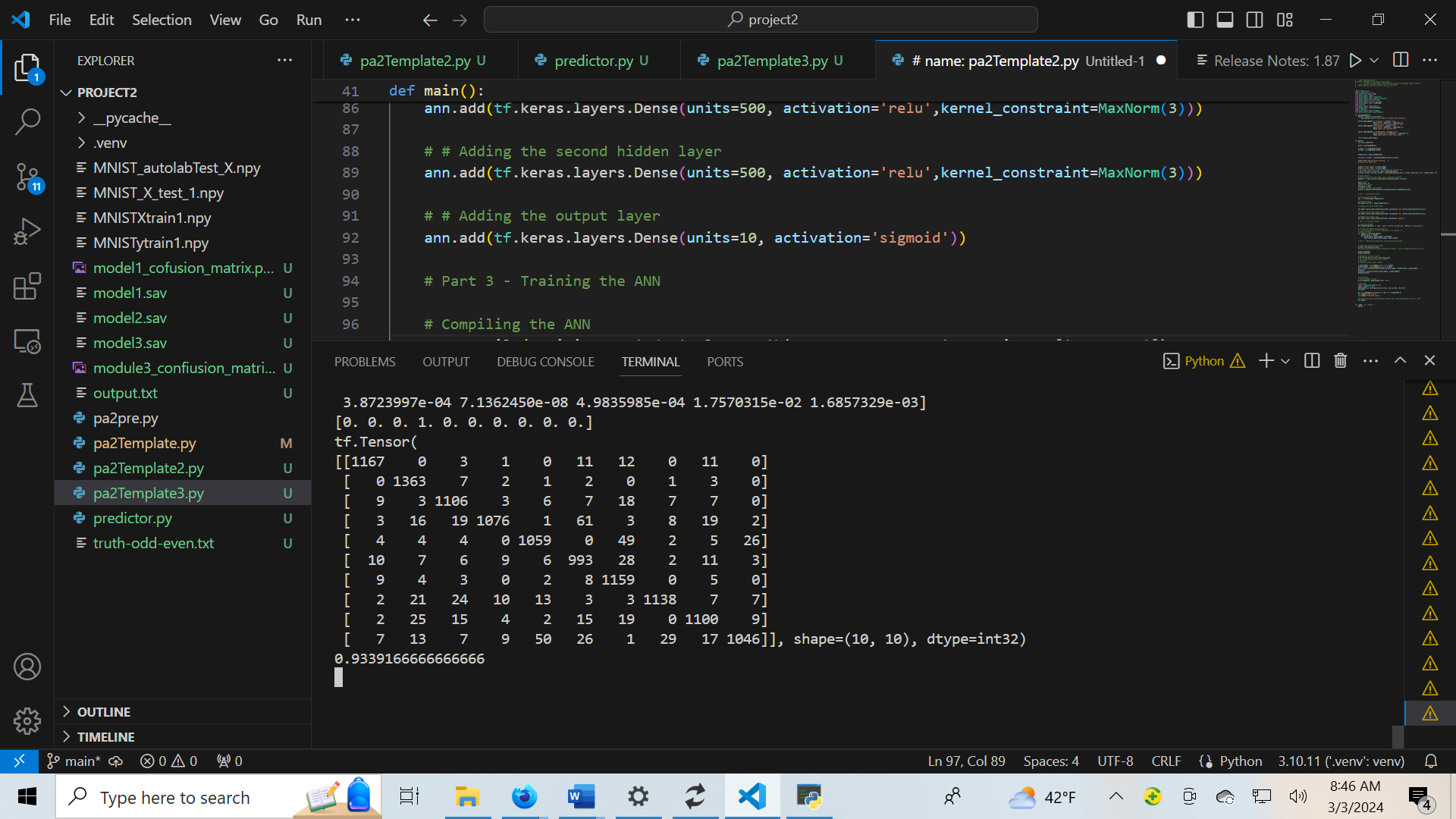
https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

 .\pa2Template2.py --training\_x .\drive\Mydrive\CS584\project2\MNISTXtrain1.npy --training\_y .\drive\Mydrive\CS584\project2\MNISTytrain1.npy --outModelFile model1.sav

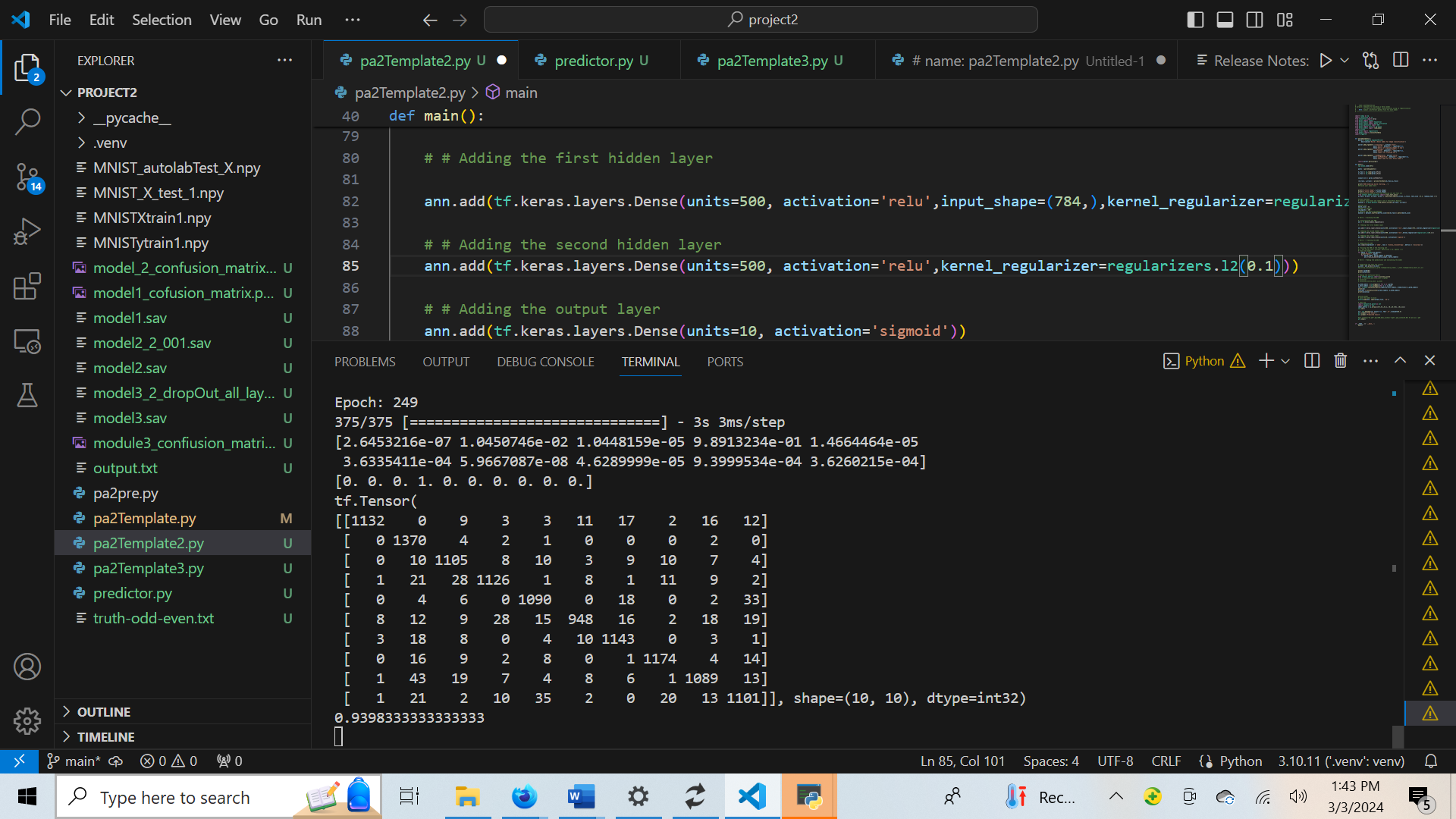
Model1



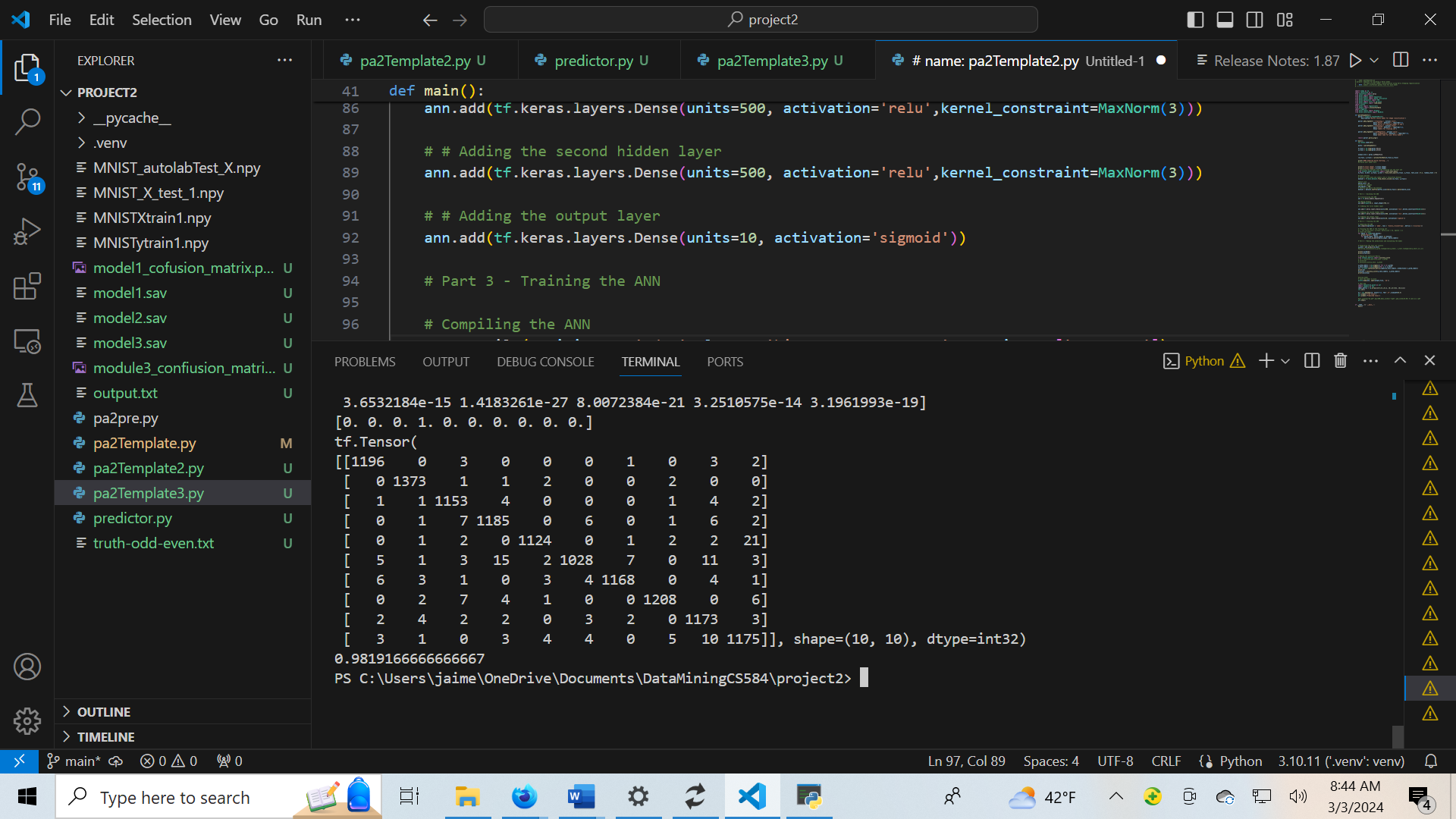
Model 2



Model 2 using lambda= 001



Model 3



Model 3 using dropout on every layer

