Forward Feed Neural Network:

- Create 5 FFNN models that will classify the 5 target variables
- Split dataset into training and test:
    - 70 training, 15% validation, 15% test. The importance of this is that the test set should not be a part of the learning process. Otherwise overfitting occurs.
- Tuning parameters:
    - Number of "Hidden Layers"
    - Number of "Neurons"
    - Combination of different "Neurons" with "Hidden Layers"
    - Optimizer
    - Loss
- Number of "Hidden Layers":
    - Tested out a range of "Hidden Layers" from 1 – 5.
        - Found that although not much difference occurs, and complexity isn't that large. Best performing was 2 layers and 5 layers.
            - Tested both a little more, stuck with 2 layers as it did the job and the complexity is less.
- Number of "Neurons"
    - Again, tested with a range, scaling up by power of 2 ([64, 128, 256, 512]), with the exception of 36 "Neurons" included as this was the size of the input features.
    - Found 36 "Neurons" in each layer to work well.
- Optimizer:
    - Tested out SGD, Adam and RMSprop. Adam optimizer worked out really well.
    - Had issues with loss being "nan" initially. Dropping learning rate helped.
- Loss:
    - Tested out a few "Loss" variables, sparse_categorical_crossentropy worked really well. Didn't need to encode target variables, as although they appeared numeric, they were also categorical. Had I used "categorical_crossentropy", I may have had to encode using one-hot encoding method and extract the max value.
- Although didn't seem to have big overfitting issues, still created models with "Dropout" layers off different dropout values. Dropout value of 0.4 seemed to work, reducing what I deemed could be overfitting