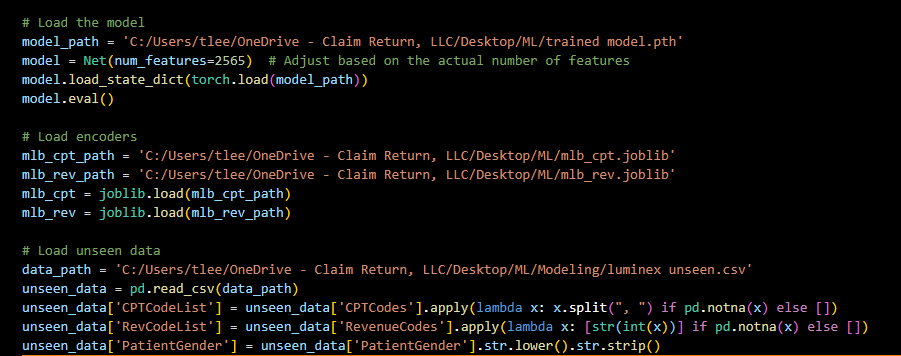
LuminX



1. Se importan las librerías necesarias: `torch` para el aprendizaje profundo, `torch.nn` para construir redes neuronales, `pandas` para manejar datos tabulares, `joblib` para cargar objetos serializados y `MultiLabelBinarizer` de scikit-learn para codificar etiquetas múltiples.

2. Se define la arquitectura de la red neuronal `Net` con tres capas densamente conectadas (fully connected layers), una capa de dropout y una función de activación sigmoidea en la capa de salida. La arquitectura tiene 2565 características de entrada y una salida binaria.



3. Se carga el modelo entrenado previamente desde un archivo `trained\_model.pth` y se coloca en modo de evaluación.

4. Se cargan los codificadores `MultiLabelBinarizer` previamente entrenados desde archivos `mlb\_cpt.joblib` y `mlb\_rev.joblib`.

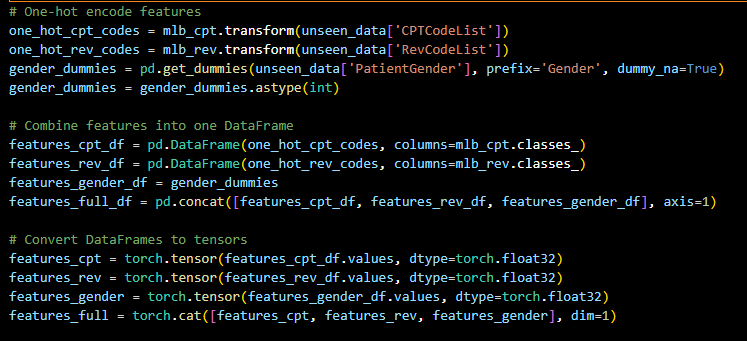
5. Se carga un conjunto de datos sin ver `unseen\_data` desde un archivo CSV.

6. Se preprocesa el conjunto de datos sin ver:

- Se dividen las cadenas de códigos CPT en listas.

- Se convierten los códigos de ingresos (revenue codes) a cadenas.

- Se convierten los géneros a minúsculas y se eliminan los espacios en blanco.



7. Se codifican las características de entrada:

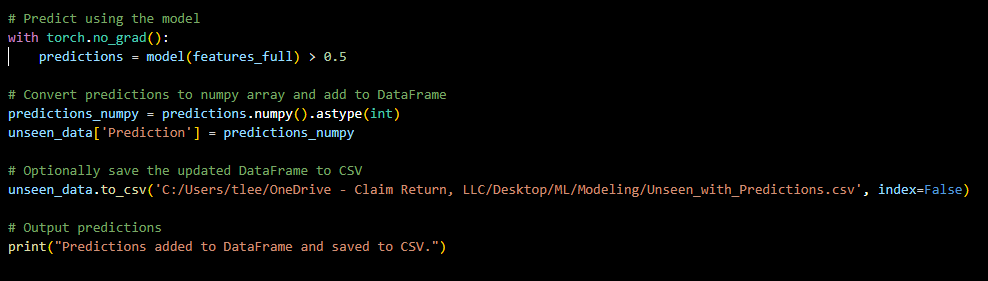
- Se aplican los codificadores `MultiLabelBinarizer` cargados a las columnas `'CPTCodeList'` y `'RevCodeList'` para obtener representaciones one-hot.

- Se crea una representación one-hot para la columna `'PatientGender'` usando `pd.get\_dummies`.

8. Se combinan las características codificadas en un solo DataFrame `features\_full\_df`.

9. Se convierten los DataFrames de características a tensores de PyTorch.

10. Se pasan los tensores de características a través del modelo cargado para obtener predicciones binarias.



11. Se convierten las predicciones a una matriz NumPy y se agregan como una nueva columna `'Prediction'` al DataFrame `unseen\_data`.

12. Opcionalmente, se guarda el DataFrame actualizado con las predicciones en un nuevo archivo CSV.

13. Se imprime un mensaje indicando que las predicciones se han agregado al DataFrame y guardado en un archivo CSV.

En resumen, este código carga un modelo entrenado previamente y los codificadores necesarios, procesa un conjunto de datos sin ver, codifica las características, pasa las características codificadas a través del modelo para obtener predicciones binarias, agrega las predicciones al conjunto de datos original y opcionalmente guarda el resultado en un archivo CSV.