Homework 1

Step 1: Data

1) How many data samples are included in the dataset?
   - There are about 3047 records of data in the dataset
2) Which problem will this dataset try to address?
   - The goal is to predict cancer mortality rate (aka TARGET_deathRate column in the dataset)
3) What is the minimum value and the maximum value in the dataset?
   - The minimum value of the column TARGET_deathRate is 59.7
   - The maximum value of the column TARGET_deathRate is 362.8
4) How many features in each data samples?
   - There are 33 features in each data sample
   - There is 1 target column TARGET_deathRate
5) Does the dataset have any missing information? E.g., missing features.
   - Yes, the dataset does have missing features. I Noticed few blanks in PctSomeCol18_24 column and PctPrivateCoverageAlone
6) What is the label of this dataset?
   - The label is “TARGET_deathRate”
7) How many data will you use for training, validation and testing?
   - I will use 70% for training, 10% validation and 20% for testing
8) What kind of data pre-processing will you use for your training dataset?
   For preparing the dataset for training, I applied the following pre-processing steps:
   A. Handling Missing Values
      - The column PctSomeCol18_24 was dropped because it contained over 2000 missing values, which would significantly reduce data quality if retained.
      - For PctEmployed16_Over and PctPrivateCoverageAlone, missing values were replaced with the mean of their respective columns, ensuring no loss of records while maintaining the overall distribution.
   B. Incompatible Columns
      - binnedInc (categorical): Initially considered one-hot encoding, but due to model compatibility issues and lack of strong predictive value, it was ultimately removed.
      - Geography (string): Dropped since it is not required for prediction and string-based identifiers are not usable by regression or neural network models.

Step 2: Model

Homework 1

Model	Test R-squared
Linear regression	0.5001
DNN-16	0.4250
DNN-30-8	0.3340
DNN-30-16-8	0.2305
DNN-30-16-8-4	0.1986

Step 3: Objective

- Used Mean Squared Error as the loss function in my code

  ```
  criterion = nn.MSELoss()
  ```

Step 4: Optimization

- Used Stochastic Gradient Descent (SGD) to train all of my models

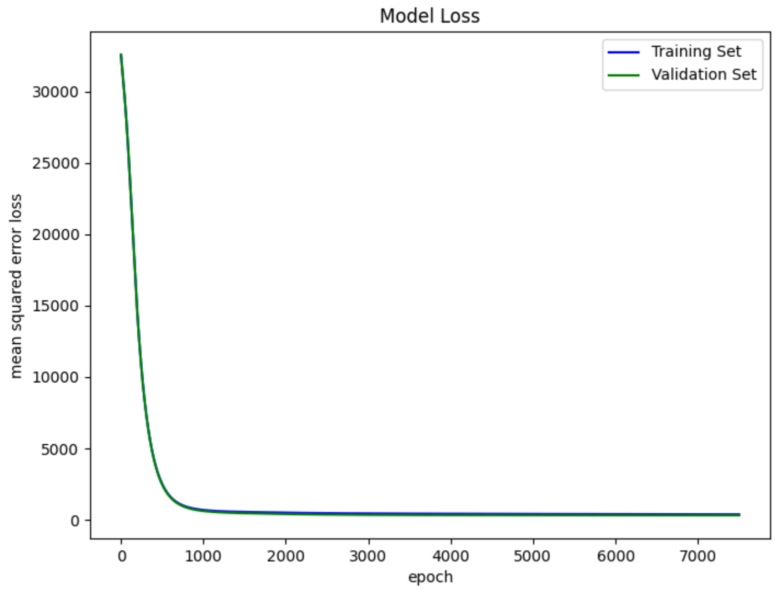
  ```
  optimizer = optim.SGD(model.parameters(), lr=learning_rate)
  ```

Step 5: Model Selection

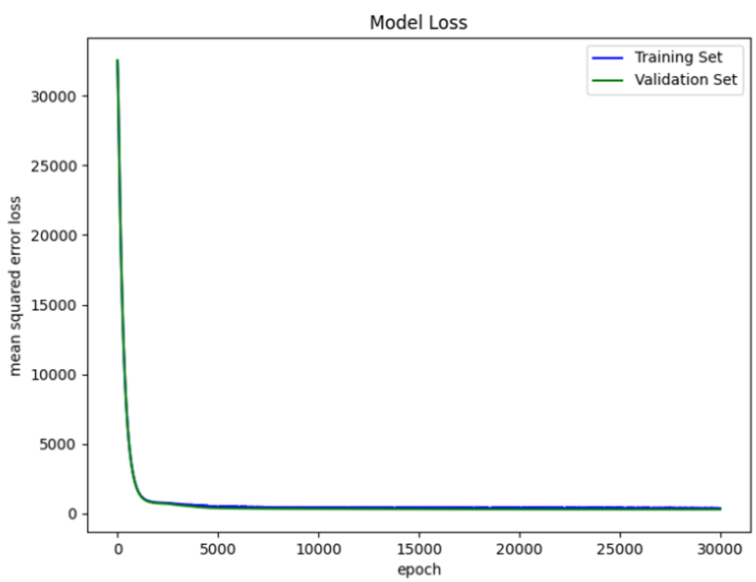
MODEL	LR 0.1	LR 0.01	LR 0.001	LR 0.0001
Linear Regression	Epoch = 700 = 0.5001	Epoch = 700 = 0.4977	Epoch = 3000 = 0.4961	Epoch = 25000 = 0.4943
DNN-16	Sigmoid Epoch = 350 = 0.4250	Sigmoid Epoch = 1000 = 0.5058	Sigmoid Epoch = 6000 = 0.5195	Sigmoid Epoch = 7500 = 0.4614
DNN-30-8	Sigmoid Epoch = 180 = 0.3376	Sigmoid Epoch = 900 = 0.5121	Sigmoid Epoch = 3800 = 0.5031	Sigmoid Epoch = 30000 = 0.4936
DNN-30-16-8	Tanh Epoch = 200 = 0.2306	Sigmoid Epoch = 3000 = 0.5346	Sigmoid Epoch = 4000 = 0.4614	Sigmoid Epoch = 3000 = 0.4603
DNN-30-16-8-4	Tanh Epoch = 90 = 0.1985	Tanh Epoch = 500 = 0.4251	Tanh Epoch = 2500 = 0.4920	Tanh Epoch = 20000 = 0.4573

Step 6: Model Performance

- Model_DN_16, LR_0.0001

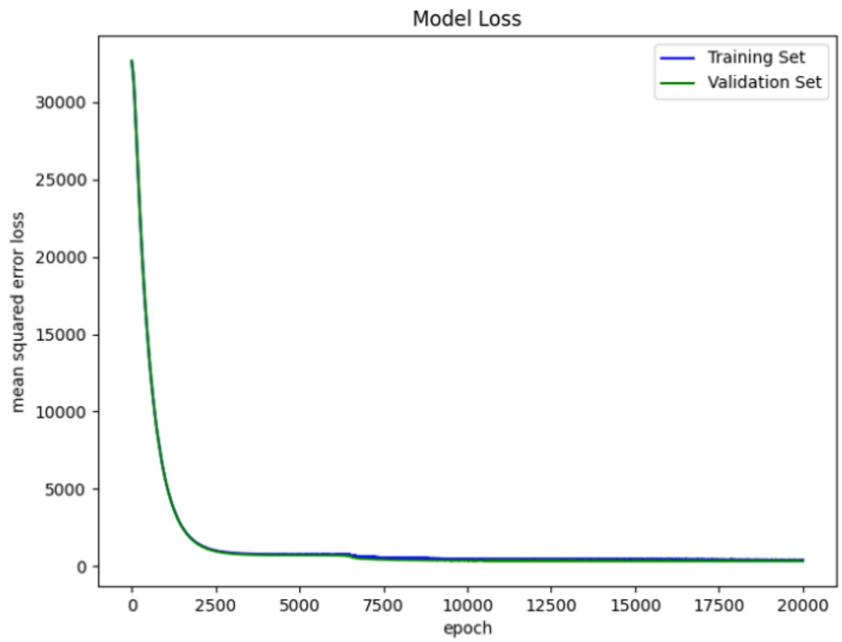


- Model_DN_30_8, LR_0.0001

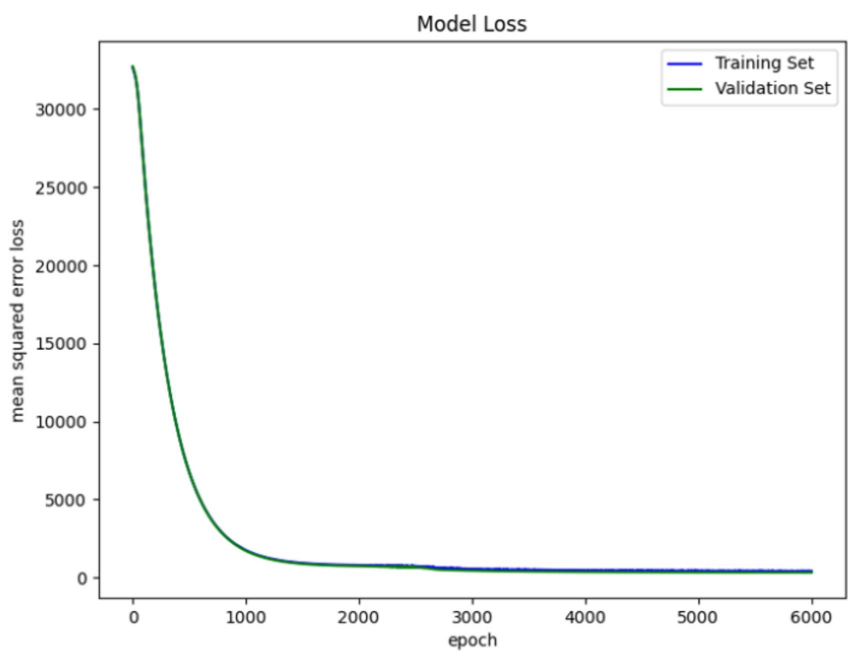


- Model_DN_30_16_8_4, LR_0.0001

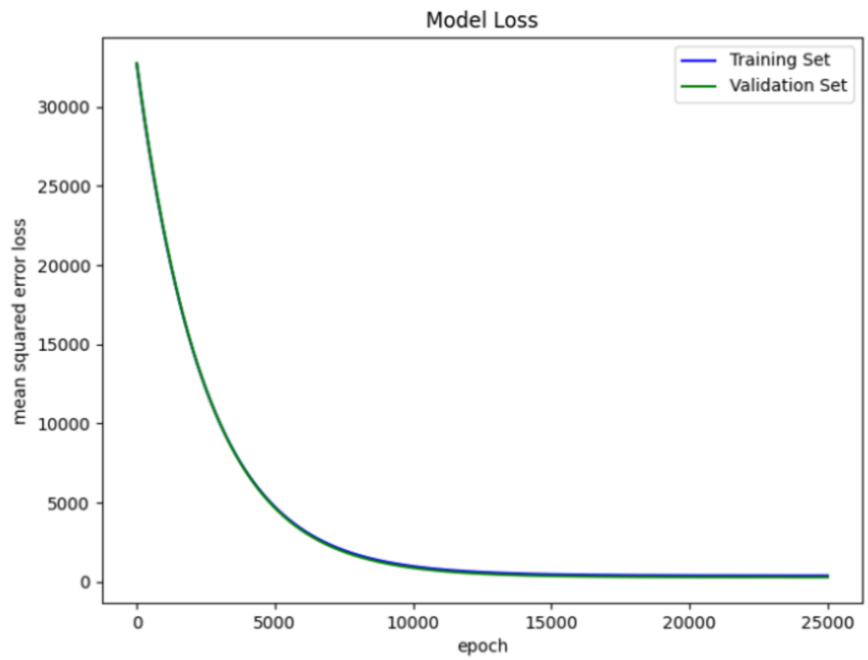
Homework 1



- Model_DN_30_16_8, LR_0.0001



- Model_Linear Regression, LR_0.0001



NOTE:

REST OF THE MODEL PERFORMANCE PLOTS ARE SAVED IN THE IMAGE FOLDER OF ZIP FILE PROVIDED