Homework 1: Linear Regression and Neural Network Regression.

Step 1: Data

1) How many data samples are included in the dataset?

- 3048 samples with 34 columns/features.

2) Which problem will this dataset try to address?

- Predicting the Cancer Mortality Rate.

3) What is the minimum value and the maximum value in the dataset?

- Maximum Value = 10170292 (popEst2015) and Minimum Value = 0

4) How many features in each data samples?

- 34

5) Does the dataset have any missing information? E.g., missing features.

- No

6) What is the label of this dataset?

- Y = TARGET_deathRate

7) How many percent of data will you use for training, validation and testing?

- Training= 80% Testing=20% , in Training Dataset we again have Training=70% and Validation=30%

8) What kind of data pre-processing will you use for your training dataset?

- Dropping unnecessary features to clean up the raw dataset.
- Imputing the NaN values in the dataset with its mean.

Step2: Model

Model	Test R-squared
Linear regression	0.502
DNN-16	-0.293
DNN-30-8	0.488
DNN-30-16-8	0.481
DNN-30-16-8-4	0.492

1. Analyze the hypothesis your model learned in terms of bias and variance. Which modelunderfitted? Which model overfitted?

1. **Linear Regression**:
   - **R^2 value**: 0.502
   - **Analysis**:
     - The R^2 value of 0.502 indicates that the model explains about 50.2% of the variance in the data, which is moderate but not great.
     - The MSE is relatively low compared to the deep learning models, suggesting that linear regression is not highly sensitive to errors.
     - **Conclusion**: The model is likely **underfitting** (high bias, low variance) as it doesn't capture the complexity of the data very well but shows a stable and low error.
2. **DNN-16**:
   - **R^2 value**: -0.293
   - **Analysis**:
     - A negative R^2 value (-0.293) suggests the model is performing worse than a simple horizontal line (mean of the data), indicating it is far from capturing the data patterns.
     - The high MSE of 1058.199 shows that the model has large errors.
     - **Conclusion**: This model is likely **underfitting** (high bias, low variance). It's too simple (with only one hidden layer of 16 units) to capture the complexity of the data.
3. **DNN-30-8**:
   - **R^2 value**: 0.488
   - **Analysis**:
     - The R^2 value of 0.488 is close to that of the linear regression model (0.502), meaning it captures about the same amount of data variance.
     - The MSE of 418.808 is slightly higher than the linear regression model's MSE.
     - **Conclusion**: This model might be **underfitting** slightly (moderate bias, low variance). It's slightly better at capturing complexity but still too simple to significantly outperform the linear regression model.
4. **DNN-30-16-8**:
   - **R^2 value**: 0.481
   - **Analysis**:
     - The R^2 value is similar to the previous models (linear regression and DNN-30-8), indicating that this model is still not capturing much more variance than the simpler models.
     - The MSE of 424.172 suggests that the model makes slightly larger errors compared to the linear regression model.
     - **Conclusion**: This model likely **underfits** (moderate bias, low variance). Adding more complexity with two hidden layers doesn't seem to improve the model significantly.
5. **DNN-30-16-8-4**:
   - **R^2 value**: 0.492
   - **Analysis**:
     - The R^2 value is 0.492, again very close to the previous models, meaning it's still not capturing much more complexity.

- The MSE is slightly lower than the previous DNN models but still not significantly better than the linear regression model.
- **Conclusion**: This model also **underfits** (moderate bias, low variance), despite its increased complexity with more layers.

Summary of Bias-Variance:

- **Linear Regression**: Likely **underfits** due to its simplicity (high bias, low variance).
- **DNN-16**: Clearly **underfits** the most, as it has a negative R^2 value and high MSE, suggesting it's not capturing any meaningful patterns (high bias).
- **DNN-30-8, DNN-30-16-8, DNN-30-16-8-4**: All of these models show similar performance and likely **underfit** as well. While they are more complex than the linear regression model, they are still not capturing enough variance, suggesting high bias and low variance.

Overfitting and Underfitting:

None of the models seem to overfit (low bias, high variance) based on the provided metrics. The deep learning models may not be complex enough (or not well-optimized) to capture the complexity of the data, resulting in all of them **underfitting**.

Step 3: Objective

Model	**Mean Square Error Loss**
Linear Regression	407.032
DNN-16	1058.199
DNN-30-8	418.808
DNN-30-16-8	424.172
DNN-30-16-8-4	415.231

Step 5: Model selection

Model	LR: 0.1 (R^2)	LR: 0.01 (R^2)	LR: 0.001 (R^2)	LR: 0.0001 (R^2)
Linear regression	0.502	0.502	0.502	0.502
DNN-16	0.820	0.201	-1.086	-31.929
DNN-30-8	-1.041	0.339	0.255	-5.800
DNN-30-16-8	-0.109	0.394	0.481	-2.581
DNN-30-16-8-4	-8.151	0.355	0.484	-6.368

Please also answer the following question in the report:

1. **Why is the learning rate impact the model performance? Can you find the best learning rate?**

The **learning rate** controls the step size at which the model's parameters (weights) are updated during training using gradient descent. If the learning rate is too large, the model may make large updates, causing it to overshoot the optimal solution and potentially not converge, leading to poor performance or even divergence (extremely poor results). On the other hand, if the learning rate is too small, the model will take very small steps and may either take too long to converge or get stuck in a local minimum.

For our data:

- When the learning rate is **too high (0.1)**, most models (except DNN-16) fail to perform well (e.g., very negative R^2 values for DNN-30-8 and DNN-30-16-8-4), which suggests that they overshot the optimal weights and didn't converge properly.
- When the learning rate is **too low (0.0001)**, all models perform very poorly, likely because they are making updates that are too small to meaningfully reduce the loss during training, resulting in extremely poor R^2 values.
- A **moderate learning rate (0.01 or 0.001)** tends to give more stable performance across the models.

Finding the Best Learning Rate:

From the data, we observe:

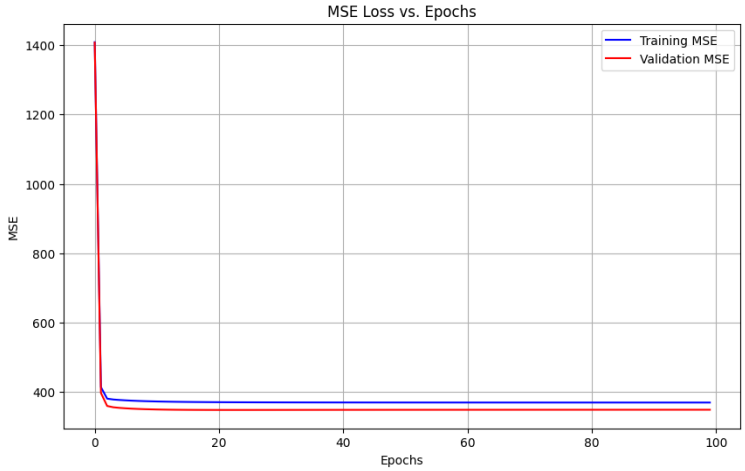
- **DNN-16** performs best with a learning rate of **0.1** ($R^2 = 0.82$), but its performance degrades with lower learning rates.
- **DNN-30-16-8-4** achieves the best performance ($R^2 = 0.484$) with a learning rate of **0.001**.
- **Linear Regression** is not impacted by learning rate since it's not trained with gradient descent, so its R^2 value remains constant (0.502).

Conclusion:

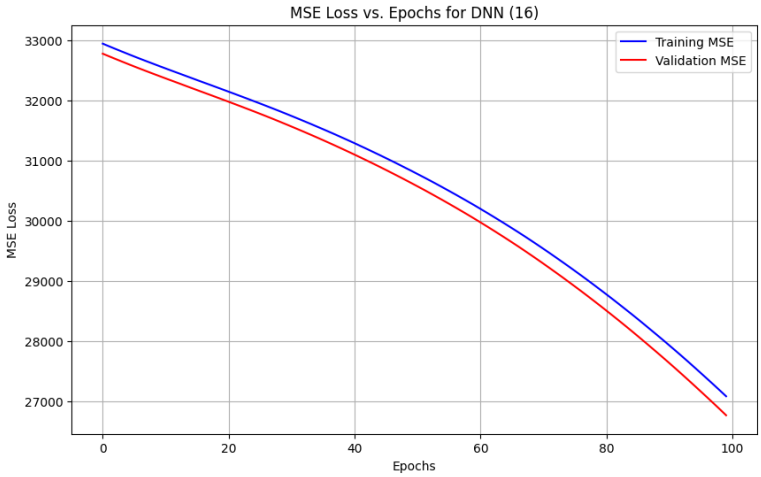
- **For DNN models**, the best learning rate seems to be **0.001**, as DNN-30-16-8-4 achieves its highest R^2 value (0.484) with this learning rate. While DNN-16 performs best at 0.1, it's generally safer to choose **0.001** as the best learning rate, since it provides stable performance across more complex models.
- **For simpler models (DNN-16)**, a higher learning rate like **0.1** can be effective.

Step 6: Model performance

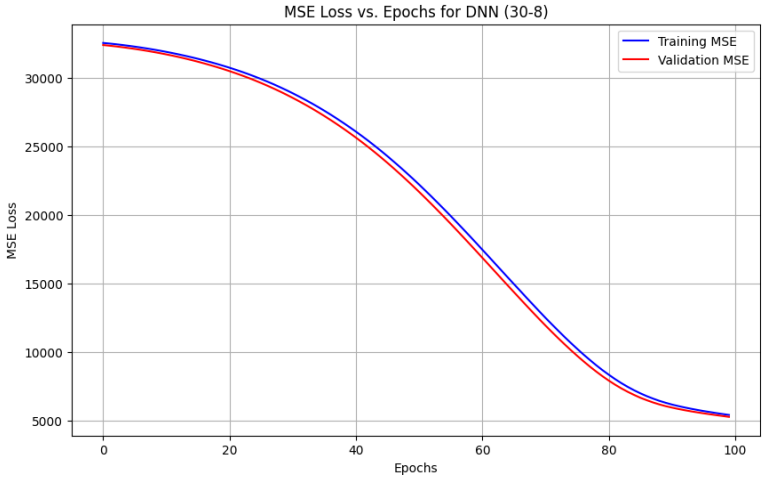
1. Linear Regression :



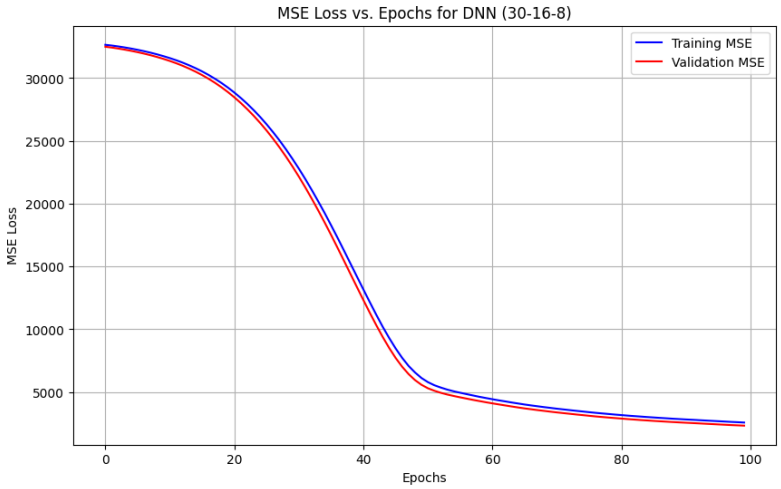
2. DNN-16 :



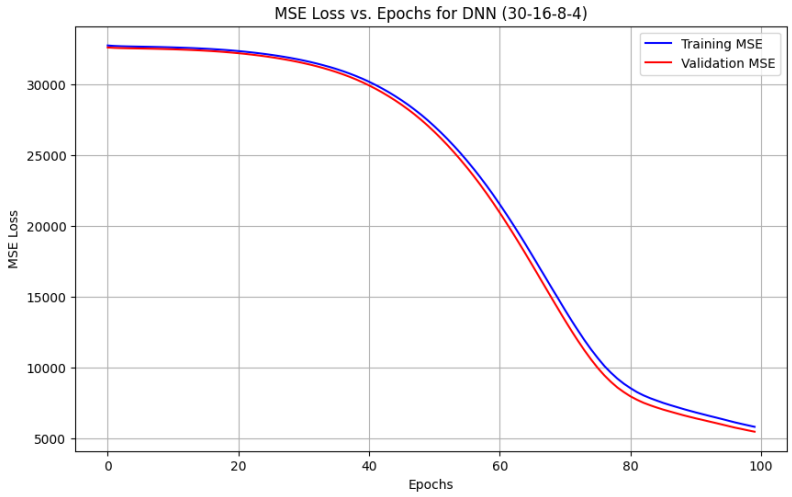
3. DNN-30-8 :



4. DNN-30-16-8 :



5. DNN-30-16-8-4 :



Highest Performing DNN –

DNN-30-16-8-4 at Learning Rate = 0.001
R-Squared = **0.484**
Mean Square Error =**415.231**