oicvjiqgk

October 5, 2024

0.1 Import the libraries

1 A

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


import warnings
warnings.filterwarnings('ignore')
```

1.1 Gathering the Data

[2]:
```
happiness_data = pd.read_csv('./data/happiness_data.csv')
happiness_data.head()
```

[2]:
```
  Country name  year  Life Ladder  Log GDP per capita  Social support  \
0  Afghanistan  2008        3.724               7.370           0.451
1  Afghanistan  2009        4.402               7.540           0.552
2  Afghanistan  2010        4.758               7.647           0.539
3  Afghanistan  2011        3.832               7.620           0.521
4  Afghanistan  2012        3.783               7.705           0.521

   Healthy life expectancy at birth  Freedom to make life choices  Generosity  \
0                             50.80                         0.718       0.168
1                             51.20                         0.679       0.190
2                             51.60                         0.600       0.121
3                             51.92                         0.496       0.162
4                             52.24                         0.531       0.236

   Perceptions of corruption  Positive affect  Negative affect
0                      0.882            0.518            0.258
1                      0.850            0.584            0.237
2                      0.707            0.618            0.275
3                      0.731            0.611            0.267
```

```
4                       0.776            0.710            0.268
```

[3]:
```
happiness_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1949 entries, 0 to 1948
Data columns (total 11 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Country name                      1949 non-null   object
 1   year                              1949 non-null   int64
 2   Life Ladder                       1949 non-null   float64
 3   Log GDP per capita                1913 non-null   float64
 4   Social support                    1936 non-null   float64
 5   Healthy life expectancy at birth  1894 non-null   float64
 6   Freedom to make life choices      1917 non-null   float64
 7   Generosity                        1860 non-null   float64
 8   Perceptions of corruption         1839 non-null   float64
 9   Positive affect                   1927 non-null   float64
 10  Negative affect                   1933 non-null   float64
dtypes: float64(9), int64(1), object(1)
memory usage: 167.6+ KB
```

In the given dataset, the feature "Country Name" is the only categorical variable, representing the names of various countries. All other features in the dataset are numeric.

In the happiness dataset you've described, the following attributes/features are continuous valued:

Log GDP per capita: Continuous because it measures the logarithm of GDP per capita.

Social support: Continuous, as it represents a numerical measure of social support.

Healthy life expectancy: Continuous, measuring the expected lifespan in years.

Freedom to make life choices: Continuous, representing a scale for how much freedom individuals feel they have.

Generosity: Continuous, as it measures the level of donations made by individuals.

Perceptions of corruption: Continuous, representing how much people perceive corruption, usually measured on a scale.

Positive affect: Continuous, representing the degree of positive emotions (happiness, laughter, etc.).

Negative affect: Continuous, representing the degree of negative emotions (worry, sadness, anger, etc.).

Life ladder: Continuous, as it measures the happiness score on a scale.

[4]:
```
happiness_data['Country name'].value_counts()
```

[4]:
```
Country name
Zimbabwe        15
```

```
South Africa    15
Tanzania        15
Denmark         15
Tajikistan      15
                ..
Maldives         1
Suriname         1
Cuba             1
Oman             1
Guyana           1
Name: count, Length: 166, dtype: int64
```

[5]:

```python
print("No of countries : ", happiness_data['Country name'].nunique())
print(happiness_data['Country name'].unique())
```

```
No of countries :  166
['Afghanistan' 'Albania' 'Algeria' 'Angola' 'Argentina' 'Armenia'
 'Australia' 'Austria' 'Azerbaijan' 'Bahrain' 'Bangladesh' 'Belarus'
 'Belgium' 'Belize' 'Benin' 'Bhutan' 'Bolivia' 'Bosnia and Herzegovina'
 'Botswana' 'Brazil' 'Bulgaria' 'Burkina Faso' 'Burundi' 'Cambodia'
 'Cameroon' 'Canada' 'Central African Republic' 'Chad' 'Chile' 'China'
 'Colombia' 'Comoros' 'Congo (Brazzaville)' 'Congo (Kinshasa)'
 'Costa Rica' 'Croatia' 'Cuba' 'Cyprus' 'Czech Republic' 'Denmark'
 'Djibouti' 'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador' 'Estonia'
 'Ethiopia' 'Finland' 'France' 'Gabon' 'Gambia' 'Georgia' 'Germany'
 'Ghana' 'Greece' 'Guatemala' 'Guinea' 'Guyana' 'Haiti' 'Honduras'
 'Hong Kong S.A.R. of China' 'Hungary' 'Iceland' 'India' 'Indonesia'
 'Iran' 'Iraq' 'Ireland' 'Israel' 'Italy' 'Ivory Coast' 'Jamaica' 'Japan'
 'Jordan' 'Kazakhstan' 'Kenya' 'Kosovo' 'Kuwait' 'Kyrgyzstan' 'Laos'
 'Latvia' 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Lithuania' 'Luxembourg'
 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta' 'Mauritania'
 'Mauritius' 'Mexico' 'Moldova' 'Mongolia' 'Montenegro' 'Morocco'
 'Mozambique' 'Myanmar' 'Namibia' 'Nepal' 'Netherlands' 'New Zealand'
 'Nicaragua' 'Niger' 'Nigeria' 'North Cyprus' 'North Macedonia' 'Norway'
 'Oman' 'Pakistan' 'Palestinian Territories' 'Panama' 'Paraguay' 'Peru'
 'Philippines' 'Poland' 'Portugal' 'Qatar' 'Romania' 'Russia' 'Rwanda'
 'Saudi Arabia' 'Senegal' 'Serbia' 'Sierra Leone' 'Singapore' 'Slovakia'
 'Slovenia' 'Somalia' 'Somaliland region' 'South Africa' 'South Korea'
 'South Sudan' 'Spain' 'Sri Lanka' 'Sudan' 'Suriname' 'Swaziland' 'Sweden'
 'Switzerland' 'Syria' 'Taiwan Province of China' 'Tajikistan' 'Tanzania'
 'Thailand' 'Togo' 'Trinidad and Tobago' 'Tunisia' 'Turkey' 'Turkmenistan'
 'Uganda' 'Ukraine' 'United Arab Emirates' 'United Kingdom'
 'United States' 'Uruguay' 'Uzbekistan' 'Venezuela' 'Vietnam' 'Yemen'
 'Zambia' 'Zimbabwe']
```

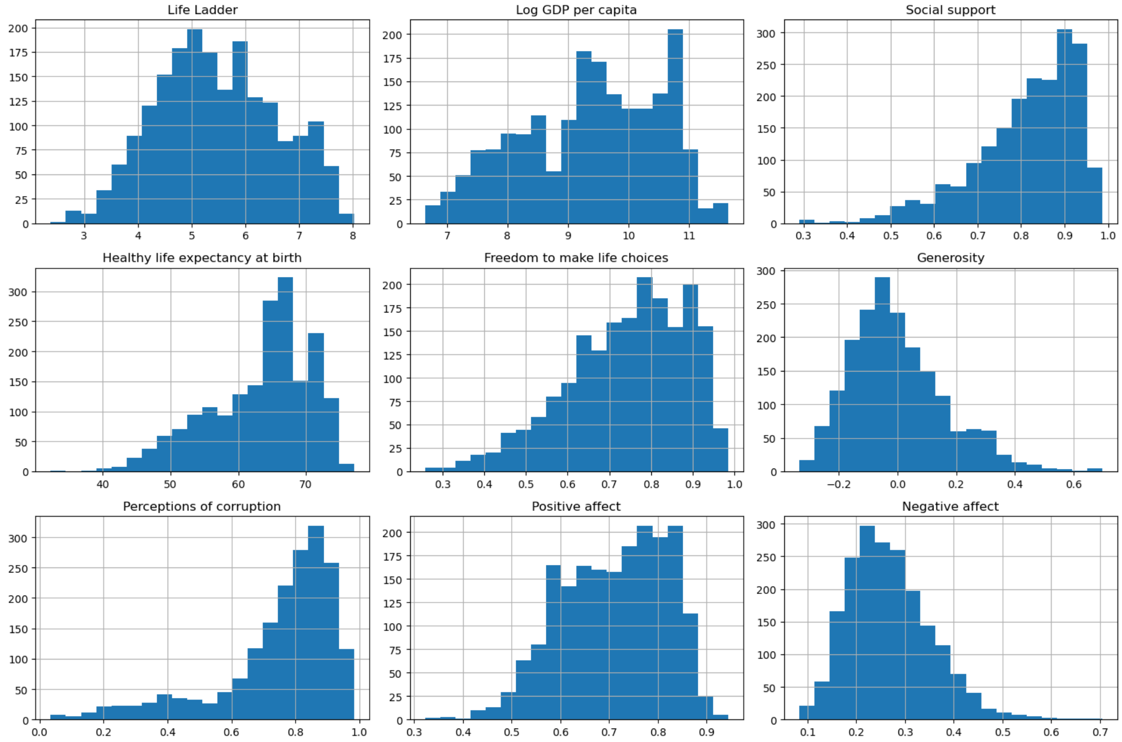
2 B

[6]:
```
happiness_data.describe()
```

[6]:
```
              year  Life Ladder  Log GDP per capita  Social support  \
count  1949.000000  1949.000000         1913.000000     1936.000000
mean   2013.216008     5.466705            9.368453        0.812552
std       4.166828     1.115711            1.154084        0.118482
min    2005.000000     2.375000            6.635000        0.290000
25%    2010.000000     4.640000            8.464000        0.749750
50%    2013.000000     5.386000            9.460000        0.835500
75%    2017.000000     6.283000           10.353000        0.905000
max    2020.000000     8.019000           11.648000        0.987000

       Healthy life expectancy at birth  Freedom to make life choices  \
count                       1894.000000                   1917.000000
mean                          63.359374                      0.742558
std                            7.510245                      0.142093
min                           32.300000                      0.258000
25%                           58.685000                      0.647000
50%                           65.200000                      0.763000
75%                           68.590000                      0.856000
max                           77.100000                      0.985000

        Generosity  Perceptions of corruption  Positive affect  \
count  1860.000000                1839.000000      1927.000000
mean      0.000103                   0.747125         0.710003
std       0.162215                   0.186789         0.107100
min      -0.335000                   0.035000         0.322000
25%      -0.113000                   0.690000         0.625500
50%      -0.025500                   0.802000         0.722000
75%       0.091000                   0.872000         0.799000
max       0.698000                   0.983000         0.944000

       Negative affect
count      1933.000000
mean          0.268544
std           0.085168
min           0.083000
25%           0.206000
50%           0.258000
75%           0.320000
max           0.705000
```

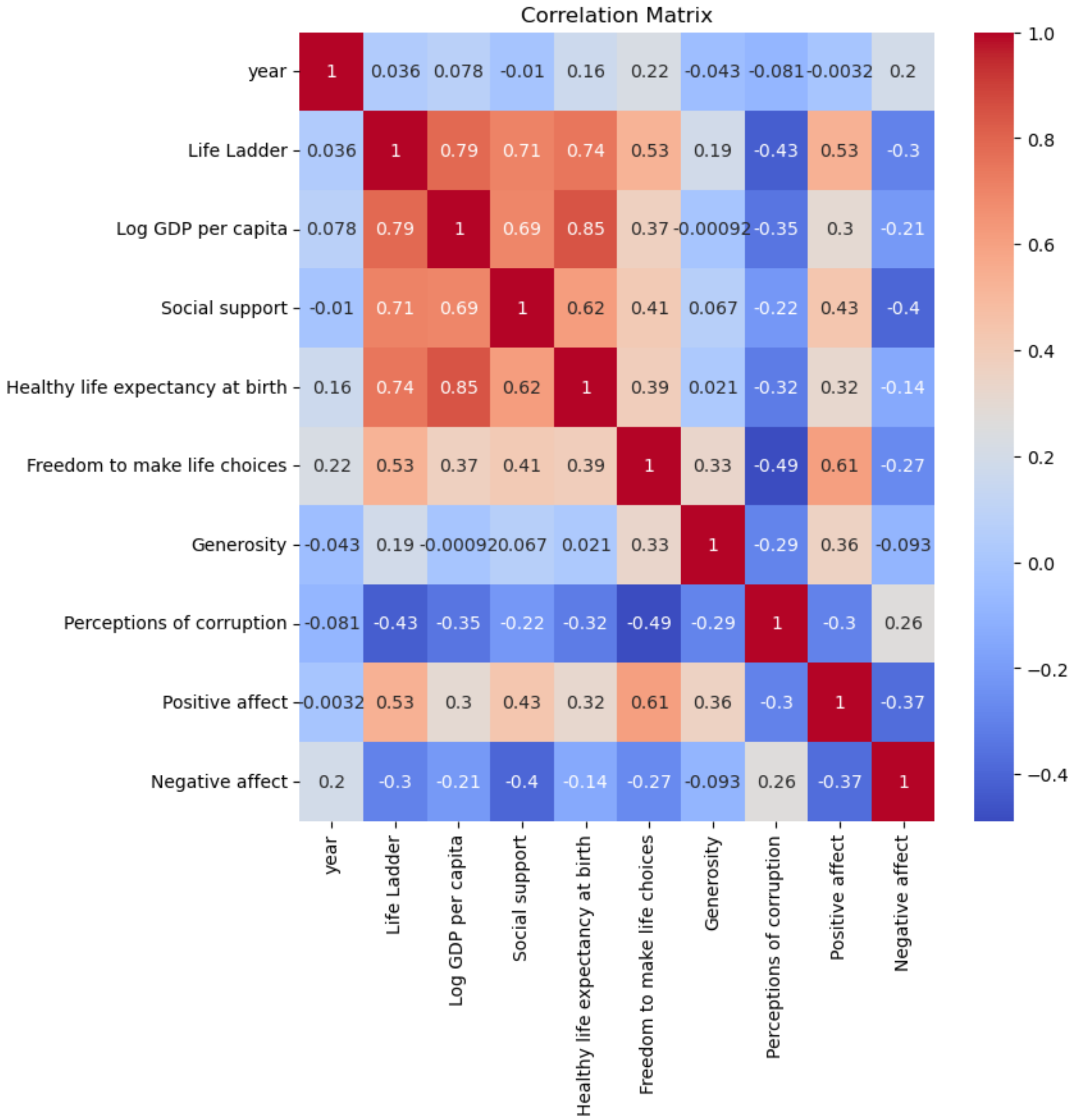
2.1 Exploratory Data Analysis

[7]:
```
numerical_cols = ['Life Ladder', 'Log GDP per capita','Social support',␣
 ↪'Healthy life expectancy at birth','Freedom to make life choices',
                   'Generosity','Perceptions of corruption', 'Positive affect',␣
 ↪'Negative affect']

happiness_data[numerical_cols].hist(figsize=(15, 10), bins=20)
plt.tight_layout()
plt.show()
```



3 C

[8]:
```
# Correlation between features

numeric_df = happiness_data.drop(['Country name'], axis=1)
correlation_matrix = numeric_df.corr()

plt.figure(figsize=(8,8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



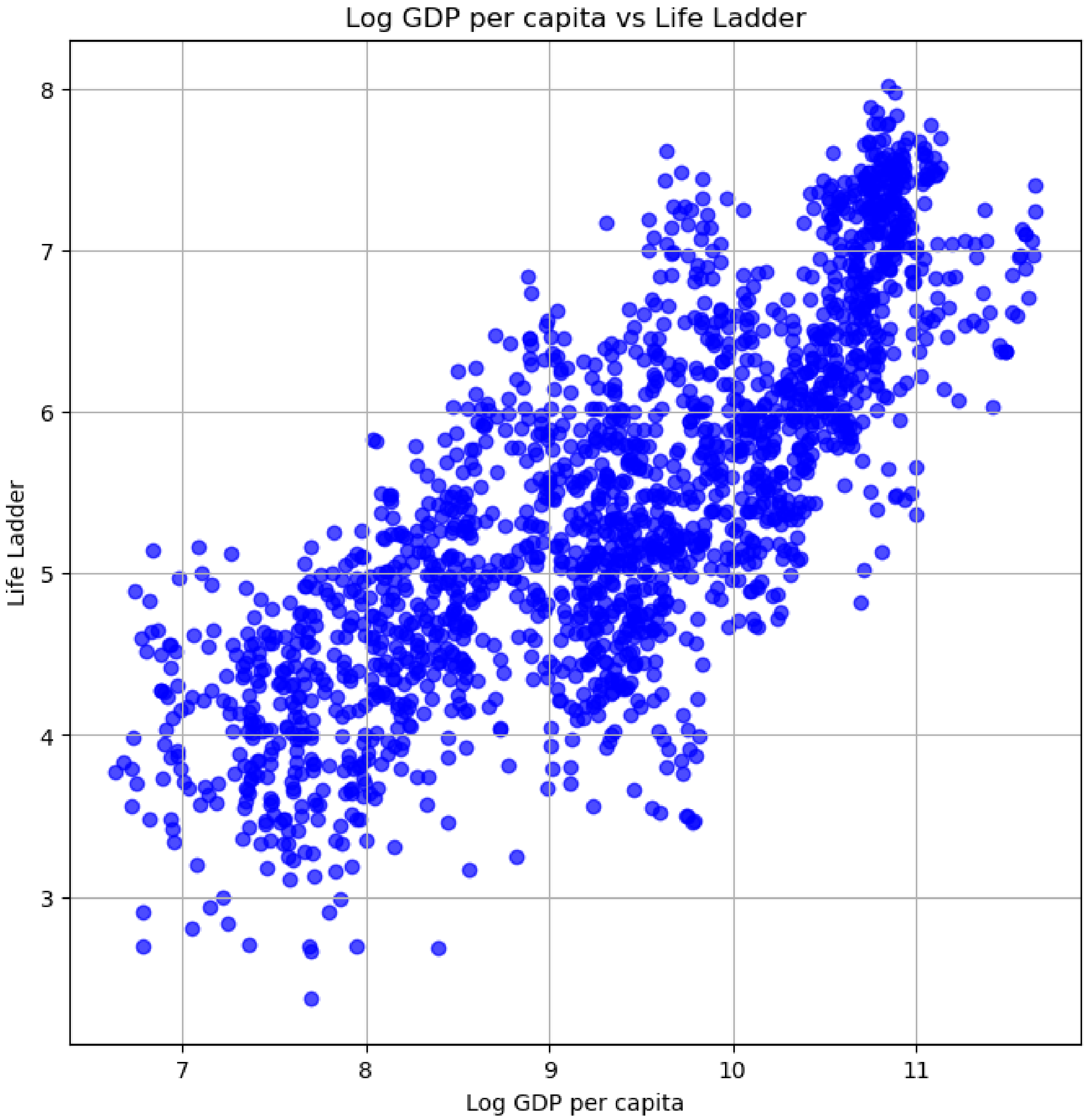
```
[9]: correlation_matrix['Life Ladder'].sort_values(ascending=False)
```

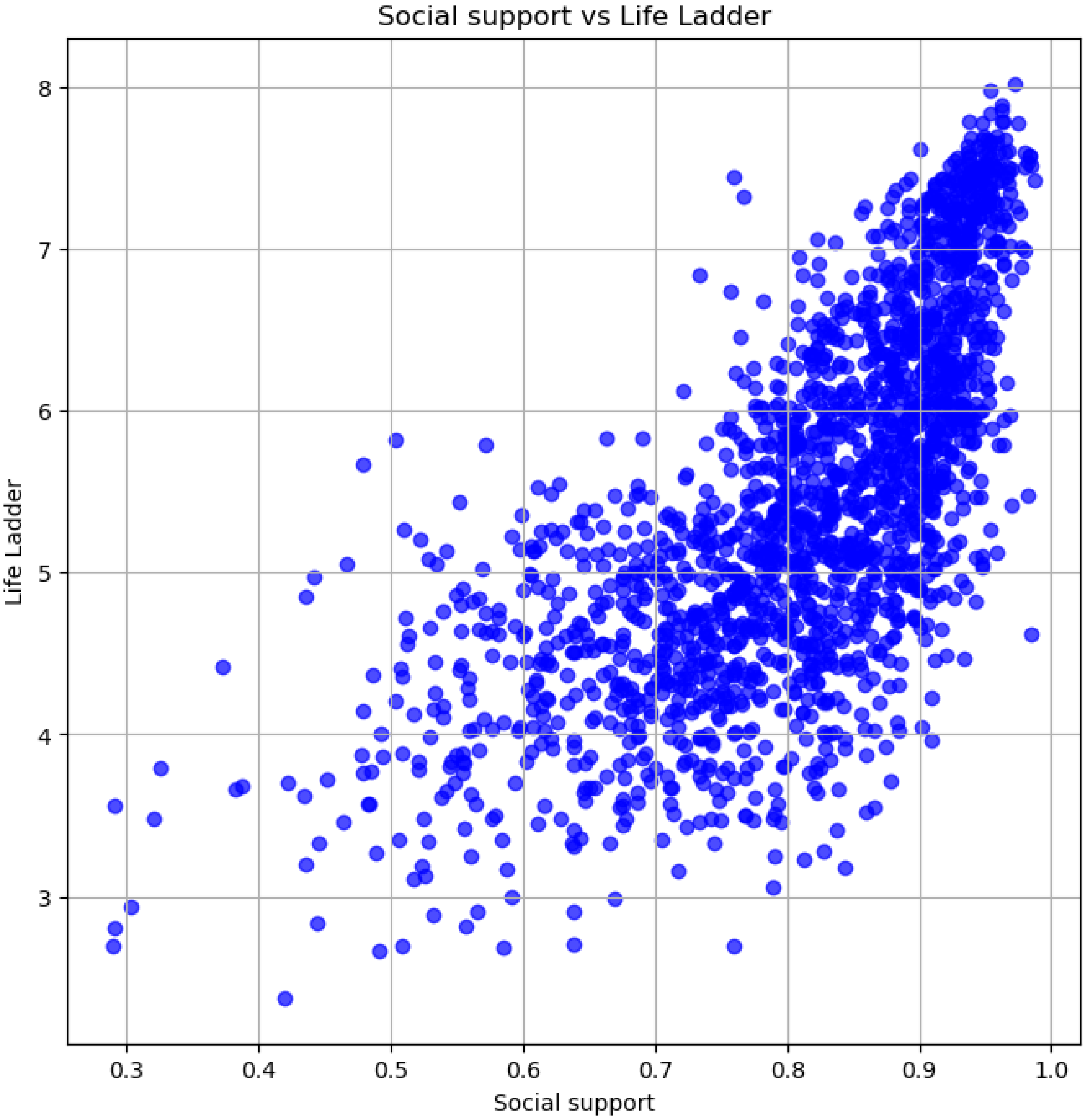
```
[9]: Life Ladder                         1.000000
     Log GDP per capita                  0.790166
     Healthy life expectancy at birth    0.744506
     Social support                      0.707806
     Positive affect                     0.532273
     Freedom to make life choices        0.528063
     Generosity                          0.190632
     year                                0.035515
     Negative affect                    -0.297488
     Perceptions of corruption          -0.427245
```

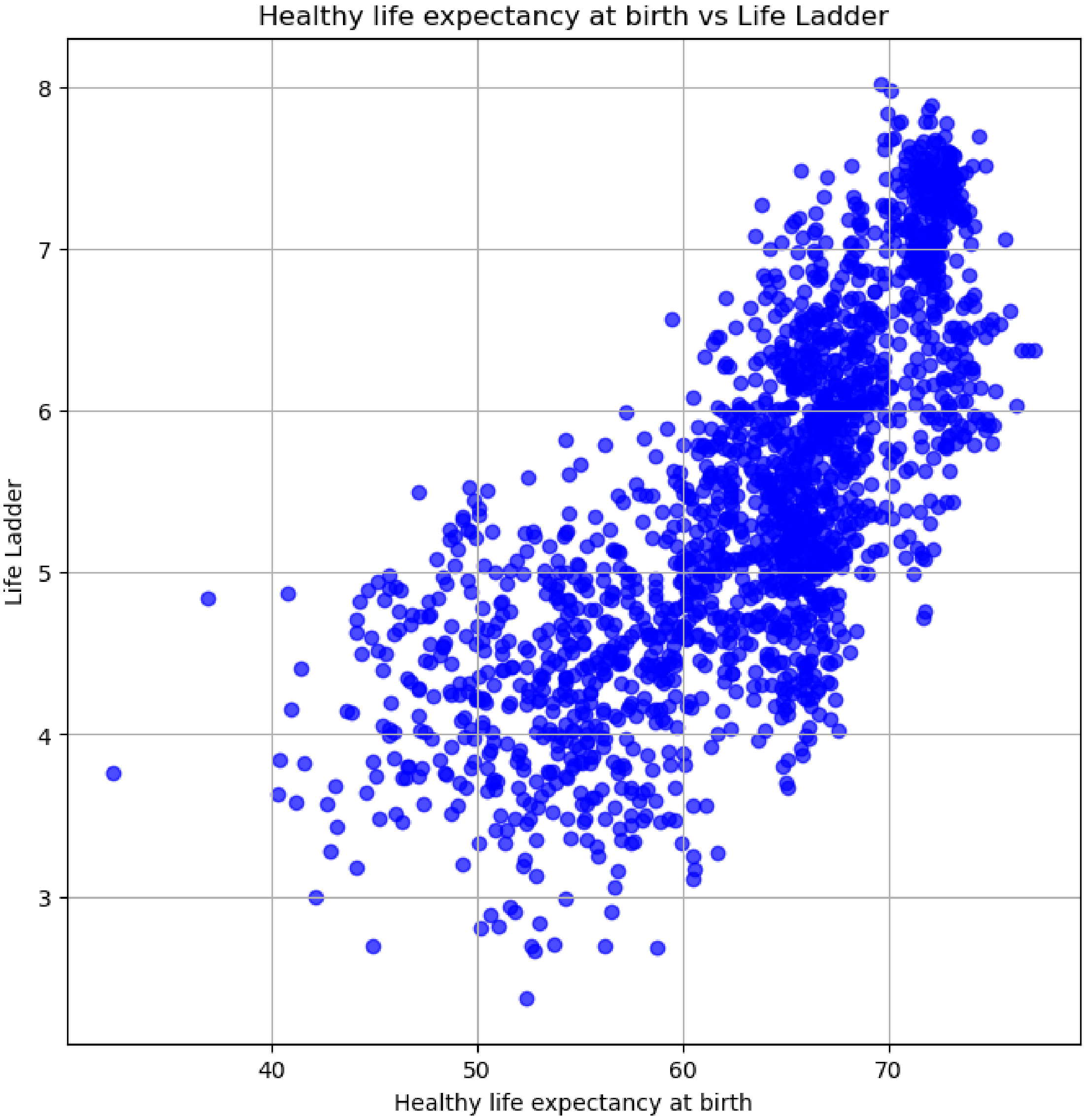
```
Name: Life Ladder, dtype: float64
```

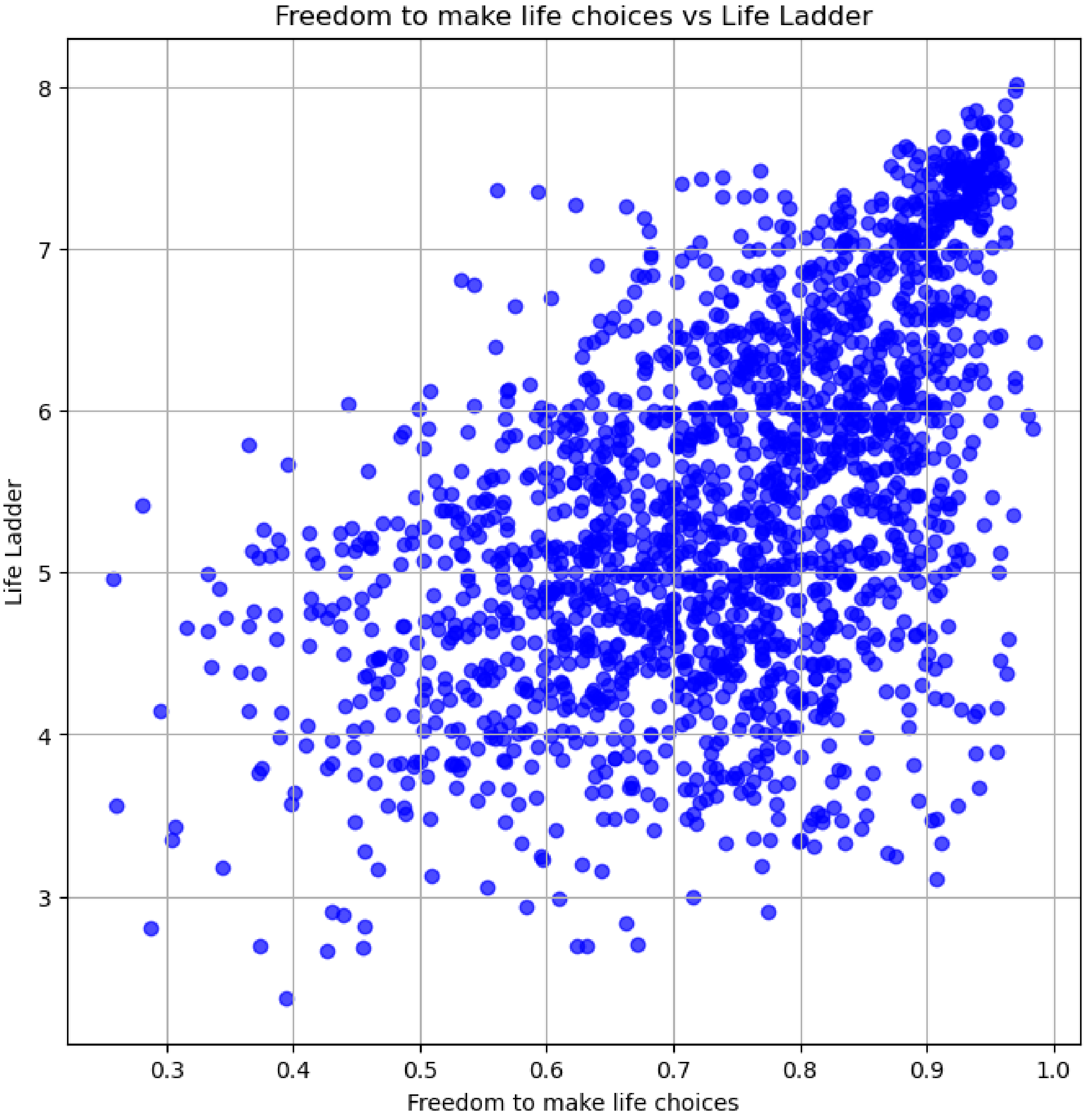
[10]:

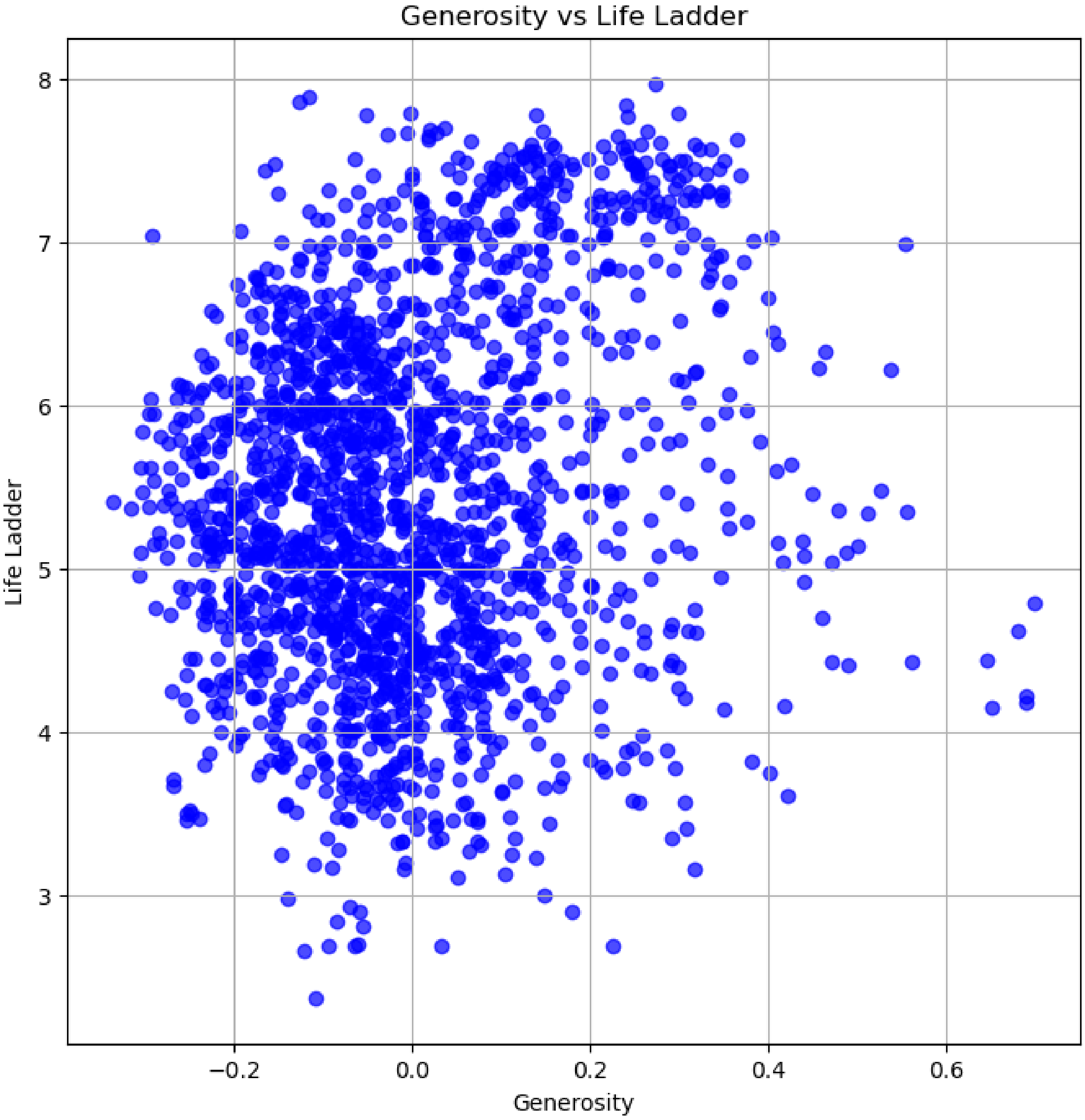
```
features = ['Log GDP per capita', 'Social support', 'Healthy life expectancy at␣
 ↪birth',
            'Freedom to make life choices', 'Generosity',
            'Perceptions of corruption', 'Positive affect', 'Negative affect']
for feature in features:
    plt.figure(figsize=(8,8))
    plt.scatter(x=happiness_data[feature], y=happiness_data['Life Ladder'],␣
 ↪color= 'blue', alpha=0.7)
    plt.title(f'{feature} vs Life Ladder')
    plt.xlabel(feature)
    plt.ylabel('Life Ladder')
    plt.grid(True)
    plt.show()
```

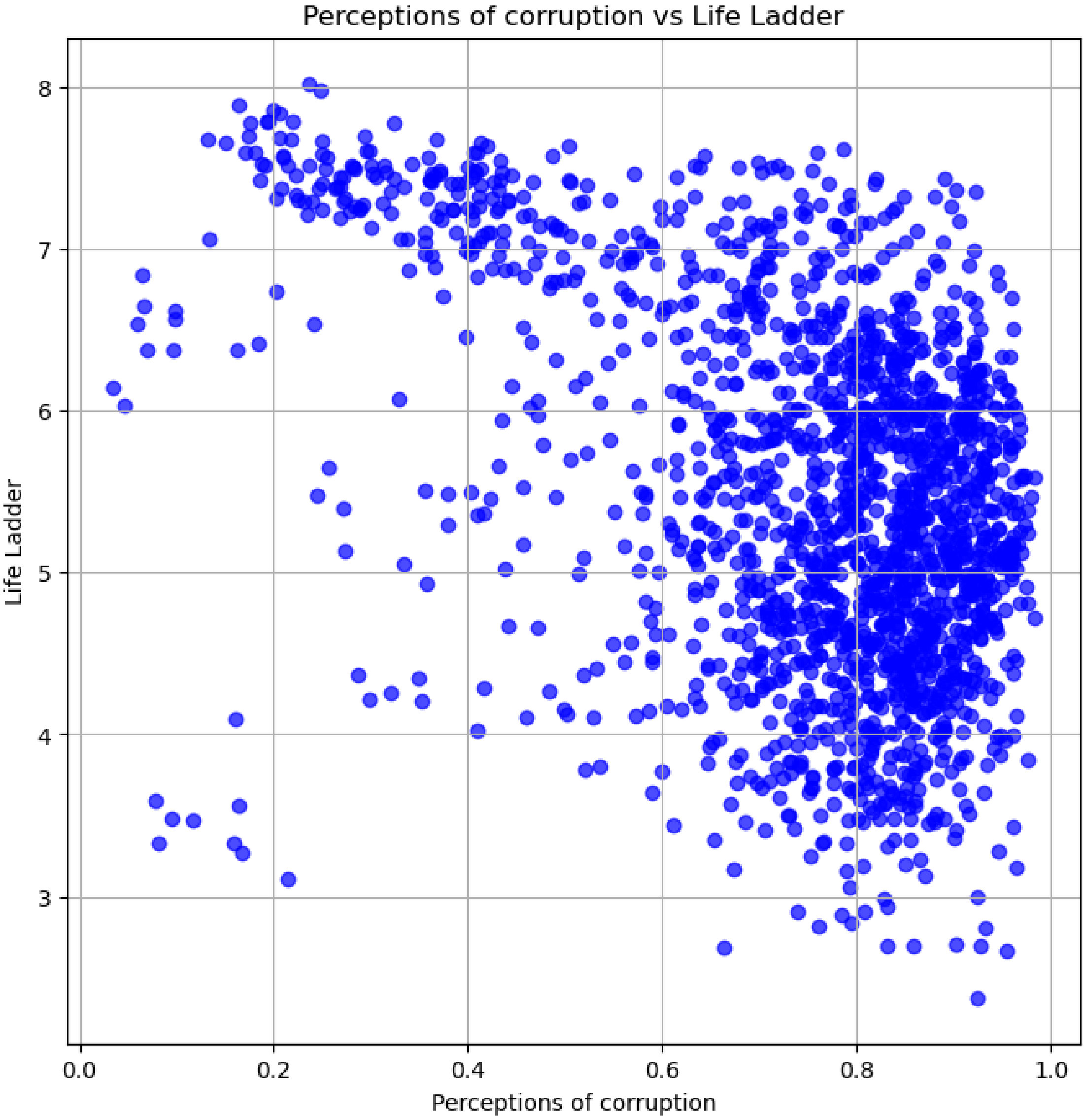

Log GDP per capita vs Life Ladder
Life Ladder
8
7
6
5
4
3
7
8
9
10
11
Log GDP per capita

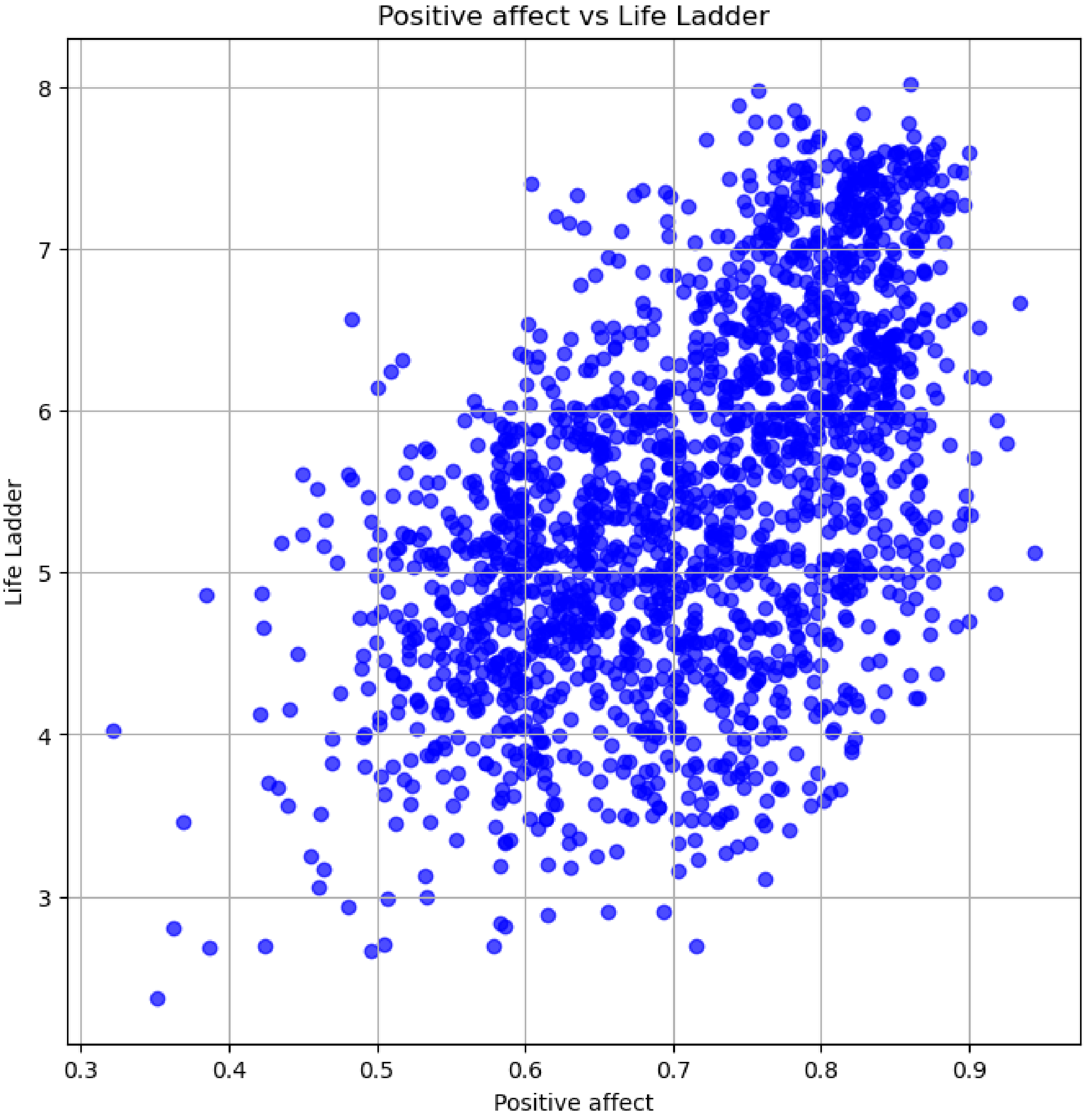

Social support vs Life Ladder
Life Ladder
8
7
6
5
4
3
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Social support

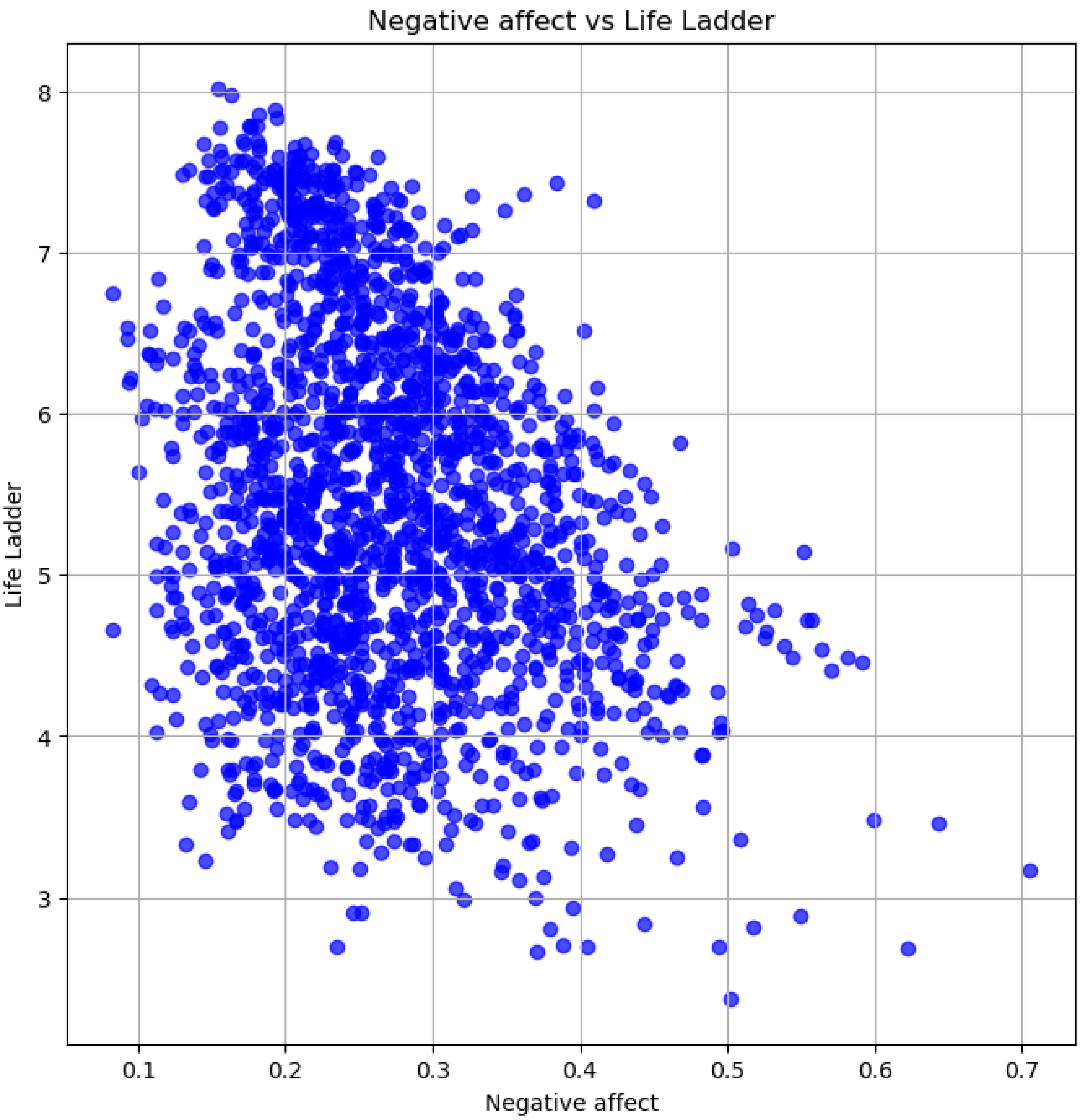

Healthy life expectancy at birth vs Life Ladder
Life Ladder
3
4
5
6
7
8
40
50
60
70
Healthy life expectancy at birth


Freedom to make life choices vs Life Ladder
Life Ladder
Freedom to make life choices
8
7
6
5
4
3
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0


Generosity vs Life Ladder
Life Ladder
Generosity
8
7
6
5
4
3
−0.2
0.0
0.2
0.4
0.6


Perceptions of corruption vs Life Ladder
Life Ladder
8
7
6
5
4
3
0.0
0.2
0.4
0.6
0.8
1.0
Perceptions of corruption


Positive affect vs Life Ladder
Life Ladder
8
7
6
5
4
3
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Positive affect



The analysis reveals that Log GDP per capita, social support, healthy life expectancy, and freedom to make choices all have a positive relationship with the Life Ladder. Additionally, positive affect is positively correlated with the Life Ladder, while negative affect shows a negative correlation.

Furthermore, a negative relationship appears between the perception of corruption and the Life Ladder, indicating that a higher perception of corruption is associated with lower levels of happiness.

No significant relationship was observed between generosity and the Life Ladder.

The correlation between positive affect and negative affect is -0.37. Since this indicates a moderate correlation, it is a good idea to include both positive affect and negative affect in your model at the start. You can evaluate their contributions during the modeling phase, and if any issues with multicollinearity or performance come up, you can reassess them at that time.

We can remove the Year feature from the model, as its correlation with the Life Ladder is 0.0355, indicating a very weak relationship.

Here are the features we will be using in the model to predict the Life Ladder, which is a continuous variable:

Log GDP per capita

Social support

Healthy life expectancy at birth

Freedom to make life choices

Positive affect

Perceptions of corruption

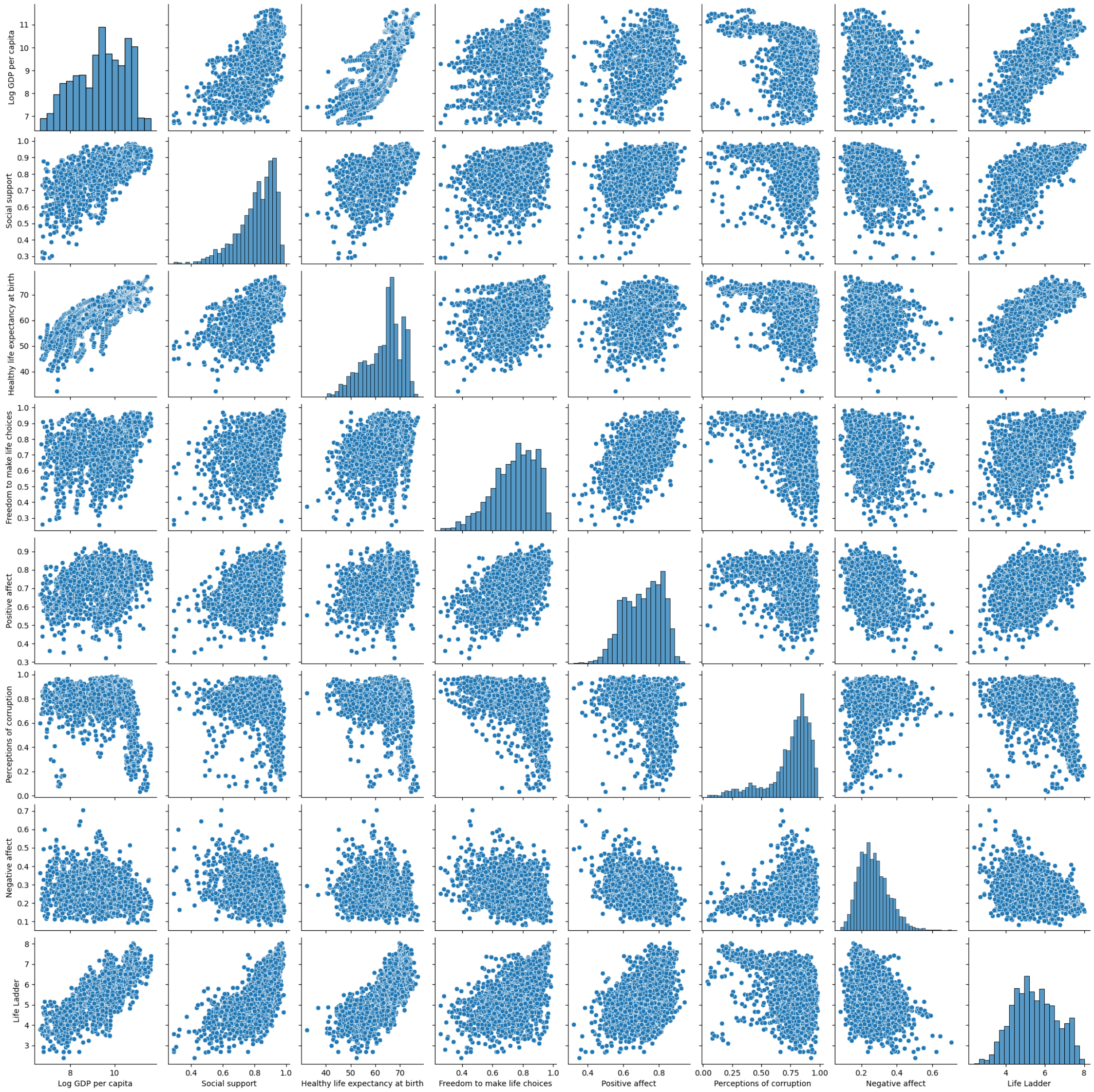
Negative affect

Generosity

Country Name

[11]:

```
# Pair Plot allowing us  to see correlations and distributions for all features␣
 ↪simultaneously.

pair_plot_features = ['Log GDP per capita', 'Social support', 'Healthy life␣
 ↪expectancy at birth',
             'Freedom to make life choices', 'Positive affect', 'Perceptions of␣
 ↪corruption',
             'Negative affect', 'Life Ladder']

sns.pairplot(happiness_data[pair_plot_features])
plt.show()
```



3.1 Handling missing values

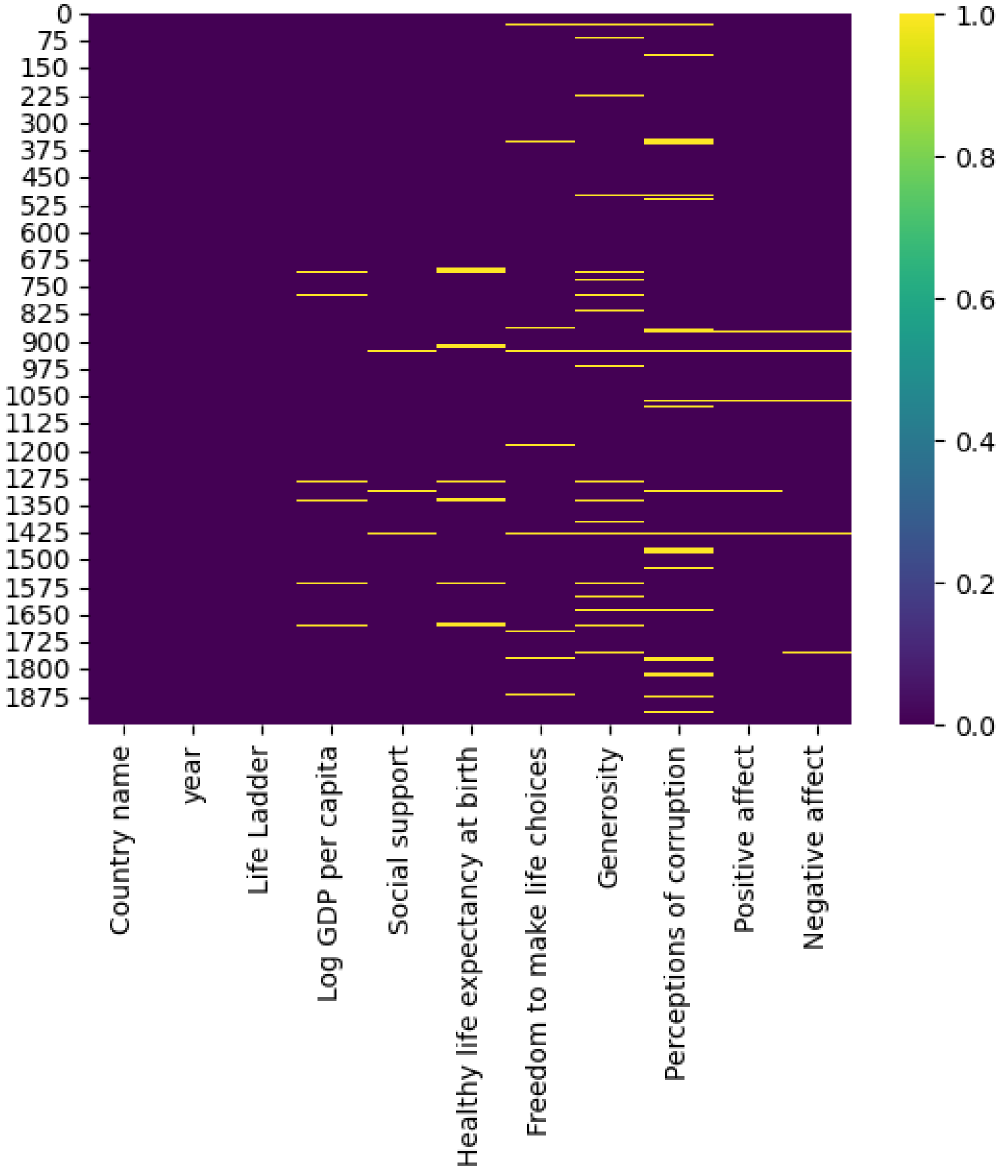
[12]:
```
happiness_data.isnull().sum()
```

[12]:
```
Country name                         0
year                                 0
Life Ladder                          0
Log GDP per capita                  36
Social support                      13
Healthy life expectancy at birth    55
Freedom to make life choices        32
Generosity                          89
```

```
Perceptions of corruption           110
Positive affect                      22
Negative affect                      16
dtype: int64
```

[13]:
```
sns.heatmap(happiness_data.isnull(), cmap='viridis')
```

[13]:
```
<Axes: >
```



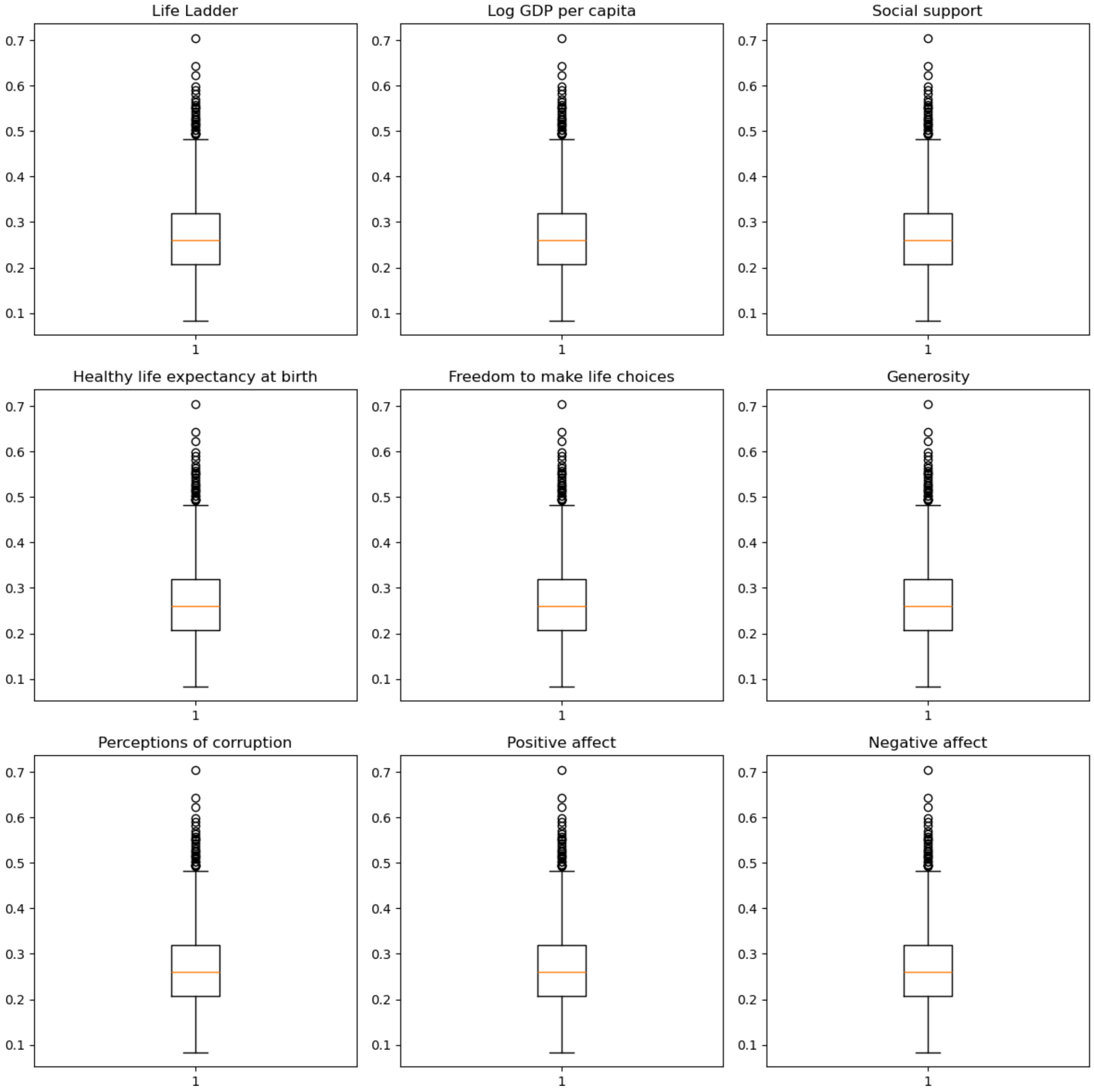
[14]:

```
missing_values_cols = ['Log GDP per capita', 'Social support', 'Healthy life␣
 ↪expectancy at birth',
                       'Freedom to make life choices', 'Generosity',␣
 ↪'Perceptions of corruption',
                       'Positive affect', 'Negative affect']

for col in missing_values_cols:
    happiness_data[col] = happiness_data[col].fillna(happiness_data[col].mean())
```

3.2 Identifying Outliers

[15]:

```
numerical_cols = ['Life Ladder', 'Log GDP per capita', 'Social support',␣
 ↪'Healthy life expectancy at birth',
                  'Freedom to make life choices', 'Generosity', 'Perceptions of␣
 ↪corruption',
                  'Positive affect', 'Negative affect']

plt.figure(figsize=(12,12))

for i, column in enumerate(numerical_cols):
    plt.subplot(3, 3, i+1)
    plt.boxplot(happiness_data[col])
    plt.title(f'{column}')

plt.tight_layout()
plt.show()
```



Outlier Assessment

After reviewing the dataset, I found that there are no outliers. The data points represent real differences in happiness across countries. Therefore, all the data is valid, and no changes are needed. This means we can use the entire dataset for analysis without removing any data points.

Categorical Encoding

Country Name is a nominal variable, one-hot encoding is the preferred method. This will prevent the model from interpreting any ordinal relationship between countries.

When dealing with a categorical variable that has a large number of categories (like 166 countries), using one-hot encoding can lead to a significant increase in the dimensionality of your dataset. This can cause issues such as increased computation time, overfitting, and difficulties in model interpretation.

Here is one of the strategy to manage this situation: Target Encoding (Mean Encoding)

[16]:
```
target_mean = happiness_data.groupby('Country name')['Life Ladder'].mean()
happiness_data['Country name encoded'] = happiness_data['Country name'].
  ↪map(target_mean)
happiness_data = happiness_data.drop(['Country name'], axis=1)
```

4 D

[17]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

idvs = ['Log GDP per capita', 'Social support','Healthy life expectancy at␣
  ↪birth', 'Freedom to make life choices',
       'Generosity', 'Perceptions of corruption', 'Positive affect','Negative␣
  ↪affect', 'Country name encoded']

dv = ['Life Ladder']

X = happiness_data[idvs]
y = happiness_data[dv]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled_df = pd.DataFrame(X_scaled, columns=idvs)

X_train, X_test, y_train, y_test = train_test_split(X_scaled_df, y, test_size=0.
  ↪25, random_state=348)

# Display the sizes of the training and testing sets
print(f'Training set size: {X_train.shape[0]}')
print(f'Testing set size: {X_test.shape[0]}')
```

```
Training set size: 1461
Testing set size: 488
```

[18]:
```
# Function to compare means and standard deviations
def compare_distributions(train, test, feature):
    print(feature)
    train_mean = train[feature].mean()
    test_mean = test[feature].mean()
    train_std = train[feature].std()
    test_std = test[feature].std()

    print(f'Feature: {feature}')
    print(f'Training Set - Mean: {train_mean:.4f}, Std: {train_std:.2f}')
    print(f'Testing Set - Mean: {test_mean:.4f}, Std: {test_std:.2f}')
    print('---')

# Compare distributions for each feature
for feature in idvs:
    compare_distributions(X_train, X_test, feature)
```

```
Log GDP per capita
Feature: Log GDP per capita
Training Set - Mean: -0.0017, Std: 1.00
Testing Set - Mean: 0.0050, Std: 1.01
---
Social support
Feature: Social support
Training Set - Mean: 0.0025, Std: 0.98
Testing Set - Mean: -0.0075, Std: 1.05
---
Healthy life expectancy at birth
Feature: Healthy life expectancy at birth
Training Set - Mean: -0.0066, Std: 1.01
Testing Set - Mean: 0.0198, Std: 0.96
---
Freedom to make life choices
Feature: Freedom to make life choices
Training Set - Mean: -0.0001, Std: 1.00
Testing Set - Mean: 0.0003, Std: 1.02
---
Generosity
Feature: Generosity
Training Set - Mean: -0.0060, Std: 1.00
Testing Set - Mean: 0.0178, Std: 0.99
---
Perceptions of corruption
Feature: Perceptions of corruption
Training Set - Mean: 0.0047, Std: 1.01
Testing Set - Mean: -0.0141, Std: 0.99
---
Positive affect
Feature: Positive affect
Training Set - Mean: 0.0180, Std: 0.99
Testing Set - Mean: -0.0540, Std: 1.02
---
Negative affect
Feature: Negative affect
Training Set - Mean: -0.0047, Std: 0.99
Testing Set - Mean: 0.0141, Std: 1.04
```

```
---
Country name encoded
Feature: Country name encoded
Training Set - Mean: 0.0004, Std: 0.99
Testing Set - Mean: -0.0013, Std: 1.03
---
```

Shuffling and Splitting

The train_test_split function randomly divides the dataset into training (75%) and testing (25%) sets while maintaining the proportion of the target variable.

Statistical Comparison

A function is created to compare the mean and standard deviation of selected numerical features between the training and testing sets. This function is called for each numerical feature to assess their similarities.

The means of the features in the training and test sets are very close to each other, indicating that both datasets are centered around the same values.The standard deviations are also very close, meaning that the variance of the features is similar between the training and testing sets.This suggests that the test data is a representative sample of the entire dataset, and therefore, the model's performance on the test set should generalize well to unseen data.

5 E

6 Linear Regression (Closed-form solution):

[19]:

```python
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score

# Initialize the Linear Regression model
model = LinearRegression()

# Perform 3-fold cross-validation with MSE as the evaluation metric
cv_scores = cross_val_score(model, X_train, y_train, cv=3,␣
 ↪scoring='neg_mean_squared_error')

# Convert the negative scores to positive MSE and compute the average MSE
mse_cv = -cv_scores.mean()
print('Cross-validation MSE:', mse_cv)
```

```
Cross-validation MSE: 0.14167399341713813
```

7 Linear Regression with Stochastic Gradient Descent (SGD):

[20]:

```python
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import SGDRegressor

# Train the model
sgd_model = SGDRegressor(max_iter=1000, tol=1e-3, learning_rate='optimal')
sgd_model.fit(X_train, y_train)

# Make predictions
y_train_pred = sgd_model.predict(X_train)
y_test_pred = sgd_model.predict(X_test)

# Calculate evaluation metrics
train_mse = mean_squared_error(y_train, y_train_pred)
test_mse = mean_squared_error(y_test, y_test_pred)
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)

# Print the results
print(f"Train MSE: {train_mse:.4f}, Train R²: {train_r2:.4f}")
print(f"Test MSE: {test_mse:.4f}, Test R²: {test_r2:.4f}")
```

```
Train MSE: 0.1722, Train R²: 0.8579
Test MSE: 0.1884, Test R²: 0.8594
```

8 Ridge Regression

[21]:

```python
from sklearn.linear_model import Ridge

# Try three different penalty (alpha) values
alpha_values = [0.1, 1.0, 10.0]

for alpha in alpha_values:
    ridge_model = Ridge(alpha=alpha)
    ridge_model.fit(X_train, y_train)

    # Predictions
    y_train_pred = ridge_model.predict(X_train)
    y_test_pred = ridge_model.predict(X_test)

    # Calculate MSE
    train_mse = mean_squared_error(y_train, y_train_pred)
    test_mse = mean_squared_error(y_test, y_test_pred)

    print(f'Ridge Regression (alpha={alpha}):')
    print(f'Train MSE: {train_mse:.4f}, Test MSE: {test_mse:.4f}')
    print('-' * 50)
```

```
Ridge Regression (alpha=0.1):
Train MSE: 0.1393, Test MSE: 0.1471
--------------------------------------------------
Ridge Regression (alpha=1.0):
Train MSE: 0.1393, Test MSE: 0.1472
--------------------------------------------------
Ridge Regression (alpha=10.0):
Train MSE: 0.1395, Test MSE: 0.1475
--------------------------------------------------
```

9 Lasso Regression

[22]:

```
from sklearn.linear_model import Lasso

# Try three different penalty (alpha) values
alpha_values = [0.001, 0.01, 0.1]

for alpha in alpha_values:
    lasso_model = Lasso(alpha=alpha)
    lasso_model.fit(X_train, y_train)

    # Predictions
    y_train_pred = lasso_model.predict(X_train)
    y_test_pred = lasso_model.predict(X_test)

    # Calculate MSE
    train_mse = mean_squared_error(y_train, y_train_pred)
    test_mse = mean_squared_error(y_test, y_test_pred)

    print(f'Lasso Regression (alpha={alpha}):')
    print(f'Train MSE: {train_mse:.4f}, Test MSE: {test_mse:.4f}')
    print('-' * 50)
```

```
Lasso Regression (alpha=0.001):
Train MSE: 0.1394, Test MSE: 0.1470
--------------------------------------------------
Lasso Regression (alpha=0.01):
Train MSE: 0.1396, Test MSE: 0.1476
--------------------------------------------------
Lasso Regression (alpha=0.1):
Train MSE: 0.1556, Test MSE: 0.1670
--------------------------------------------------
```

10 Elastic Net Regression

[23]:

```python
from sklearn.linear_model import ElasticNet

# Try three different combinations of alpha and l1_ratio
alpha_values = [0.001, 0.01, 0.1]
l1_ratio_values = [0.2, 0.5, 0.8]

for alpha in alpha_values:
    for l1_ratio in l1_ratio_values:
        elastic_net_model = ElasticNet(alpha=alpha, l1_ratio=l1_ratio)
        elastic_net_model.fit(X_train, y_train)

        # Predictions
        y_train_pred = elastic_net_model.predict(X_train)
        y_test_pred = elastic_net_model.predict(X_test)

        # Calculate MSE
        train_mse = mean_squared_error(y_train, y_train_pred)
        test_mse = mean_squared_error(y_test, y_test_pred)

        print(f'Elastic Net (alpha={alpha}, l1_ratio={l1_ratio}):')
        print(f'Train MSE: {train_mse:.4f}, Test MSE: {test_mse:.4f}')
        print('-' * 50)
```

```
Elastic Net (alpha=0.001, l1_ratio=0.2):
Train MSE: 0.1393, Test MSE: 0.1471
--------------------------------------------------
Elastic Net (alpha=0.001, l1_ratio=0.5):
Train MSE: 0.1393, Test MSE: 0.1471
--------------------------------------------------
Elastic Net (alpha=0.001, l1_ratio=0.8):
Train MSE: 0.1394, Test MSE: 0.1470
--------------------------------------------------
Elastic Net (alpha=0.01, l1_ratio=0.2):
Train MSE: 0.1396, Test MSE: 0.1476
--------------------------------------------------
Elastic Net (alpha=0.01, l1_ratio=0.5):
Train MSE: 0.1395, Test MSE: 0.1474
--------------------------------------------------
Elastic Net (alpha=0.01, l1_ratio=0.8):
Train MSE: 0.1395, Test MSE: 0.1475
--------------------------------------------------
Elastic Net (alpha=0.1, l1_ratio=0.2):
Train MSE: 0.1507, Test MSE: 0.1613
--------------------------------------------------
Elastic Net (alpha=0.1, l1_ratio=0.5):
```

```
Train MSE: 0.1488, Test MSE: 0.1599
--------------------------------------------------
Elastic Net (alpha=0.1, l1_ratio=0.8):
Train MSE: 0.1512, Test MSE: 0.1629
--------------------------------------------------
```

11 Impact of Regularization on Ridge, Lasso, and Elastic Net

11.1 1. Ridge Regression

- **Alpha values tested**: 0.1, 1.0, 10.0
- **Observation**:
  - MSE stays almost the same across different alpha values.
  - Increasing the regularization (higher alpha) barely affects performance.
  - Ridge handles regularization well, and even with stronger regularization, it doesn't show much underfitting.
- **Conclusion**:
  - Ridge is not very sensitive to changes in alpha.
  - Even with higher alpha, the model maintains good performance without much overfitting or underfitting.

11.2 2. Lasso Regression

- **Alpha values tested**: 0.001, 0.01, 0.1
- **Observation**:
  - With small alpha values (0.001), Lasso performs similarly to Ridge.
  - As alpha increases, both train and test MSE go up, indicating underfitting.
  - Lasso starts shrinking the coefficients, which hurts the model's performance for larger alpha values.
- **Conclusion**:
  - Lasso's regularization impact is more noticeable with larger alpha values, leading to underfitting as alpha increases.
  - It can also force some coefficients to zero, which is useful for feature selection but harmful if over-applied.

11.3 3. Elastic Net Regression

- **Alpha values tested**: 0.001, 0.01, 0.1 with **l1_ratio** values of 0.2, 0.5, 0.8
- **Observation**:
  - At lower alpha values (0.001 and 0.01), Elastic Net behaves like Ridge and Lasso, with no major changes across different l1_ratios.
  - However, at higher alpha (0.1), MSE increases, showing underfitting as regularization becomes too strong.
- **Conclusion**:
  - Elastic Net balances between Ridge and Lasso, but when the penalty term (alpha) is too high, it leads to underfitting, similar to Lasso.
  - The l1_ratio doesn't significantly change the results for this dataset.

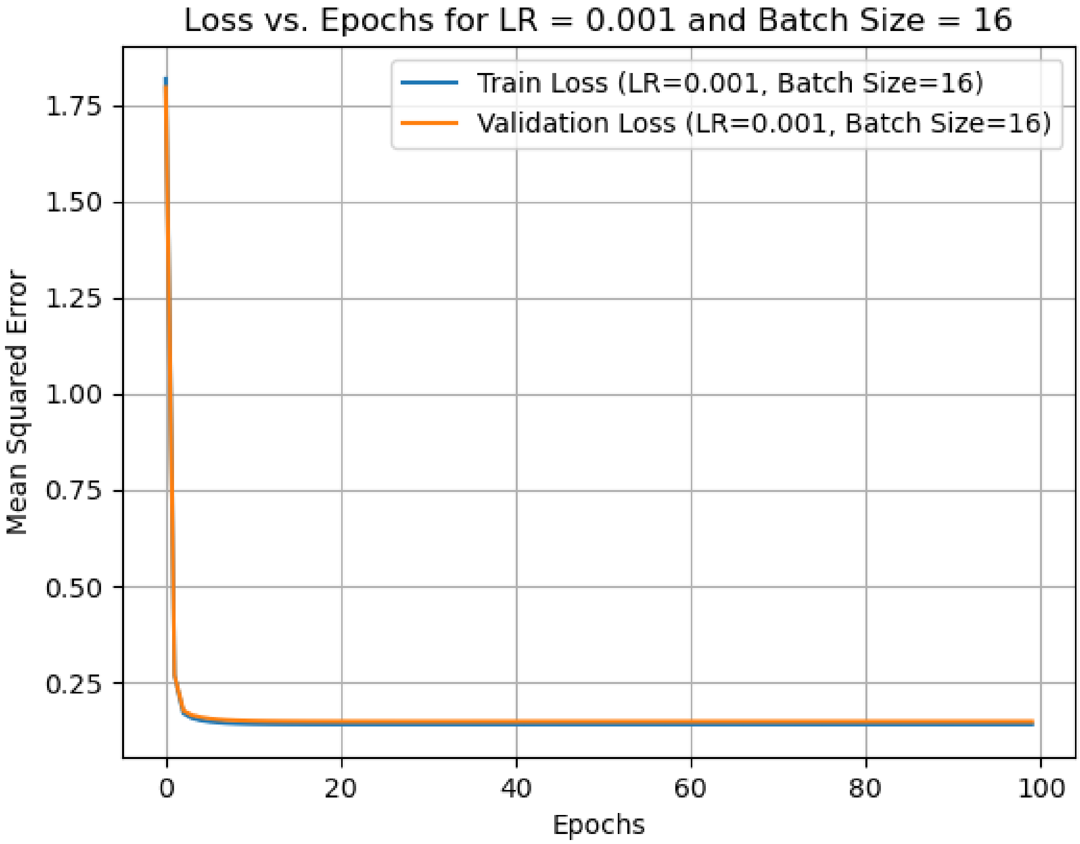
11.4 Summary

- **Ridge**: Works well even with stronger regularization and is less sensitive to changes in alpha.
- **Lasso**: Becomes more sensitive to higher regularization, leading to underfitting and coefficient reduction.
- **Elastic Net**: Combines both methods but behaves more like Lasso when the penalty is too strong.

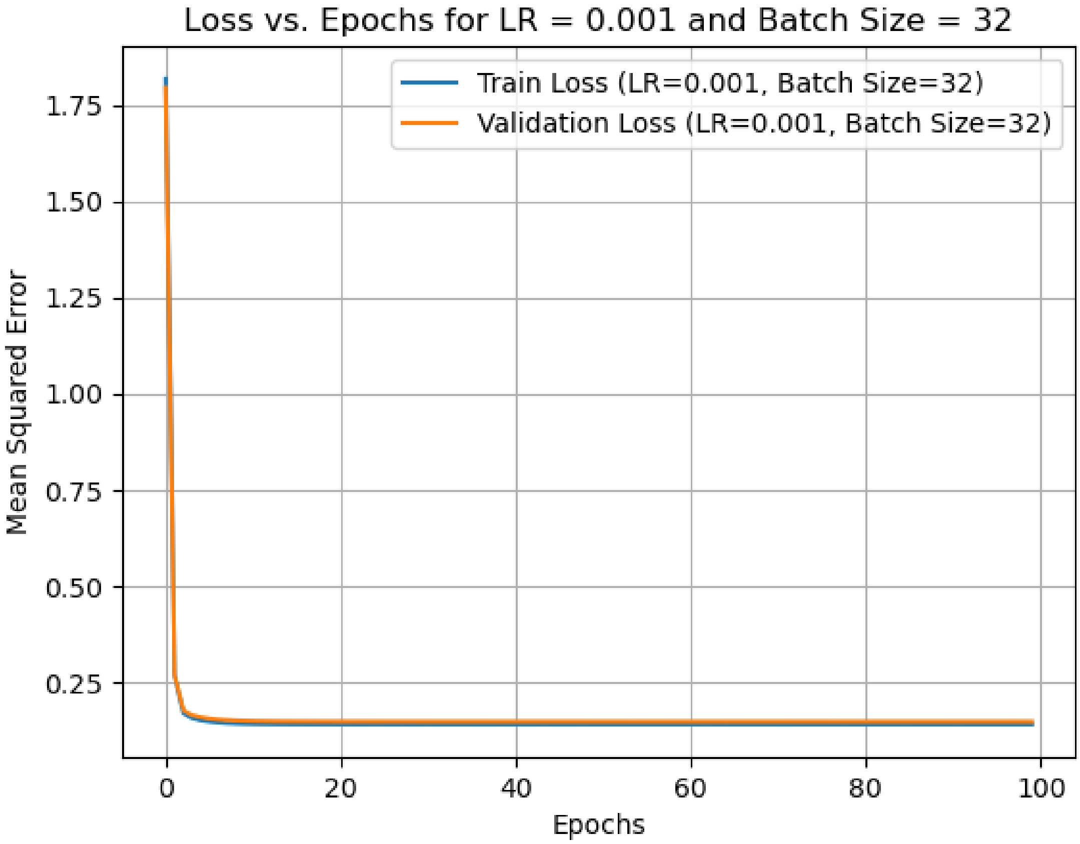
In this case, small regularization values perform best, while larger values cause underfitting.

[24]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import SGDRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split



# Define learning rates and batch sizes to test
learning_rates = [0.001, 0.01, 0.1, 0.5]
batch_sizes = [16, 32, 64]

# Store results
results = []

# Loop through each combination of learning rates and batch sizes
for lr in learning_rates:
    for batch_size in batch_sizes:
        sgd_model = SGDRegressor(max_iter=1000, tol=1e-3,␣
 ↪learning_rate='constant', eta0=lr)
        train_loss = []
        val_loss = []

        print(f"\nTraining with Learning Rate = {lr} and Batch Size =␣
 ↪{batch_size}")

        # Training loop with batching
        for epoch in range(100):
            # Shuffle and split data into batches
            for i in range(0, len(X_train), batch_size):
                X_batch = X_train[i:i+batch_size]
                y_batch = y_train[i:i+batch_size]
                sgd_model.partial_fit(X_batch, y_batch)

            # Compute training and validation loss
            train_mse = mean_squared_error(y_train, sgd_model.predict(X_train))
            val_mse = mean_squared_error(y_test, sgd_model.predict(X_test))

            train_loss.append(train_mse)
            val_loss.append(val_mse)

            # Print the losses for each epoch
            print(f"Epoch {epoch + 1}: Train MSE = {train_mse:.4f}, Validation␣
 ↪MSE = {val_mse:.4f}")
        results.append((lr, batch_size, train_loss, val_loss))
        # Create separate plots for each combination
        plt.figure()
        plt.plot(train_loss, label='Train Loss (LR={}, Batch Size={})'.
 ↪format(lr, batch_size))
        plt.plot(val_loss, label='Validation Loss (LR={}, Batch Size={})'.
 ↪format(lr, batch_size))
        plt.title('Loss vs. Epochs for LR = {} and Batch Size = {}'.format(lr,␣
 ↪batch_size))
        plt.xlabel('Epochs')
        plt.ylabel('Mean Squared Error')
        plt.legend()
        plt.grid()
        plt.show()
```

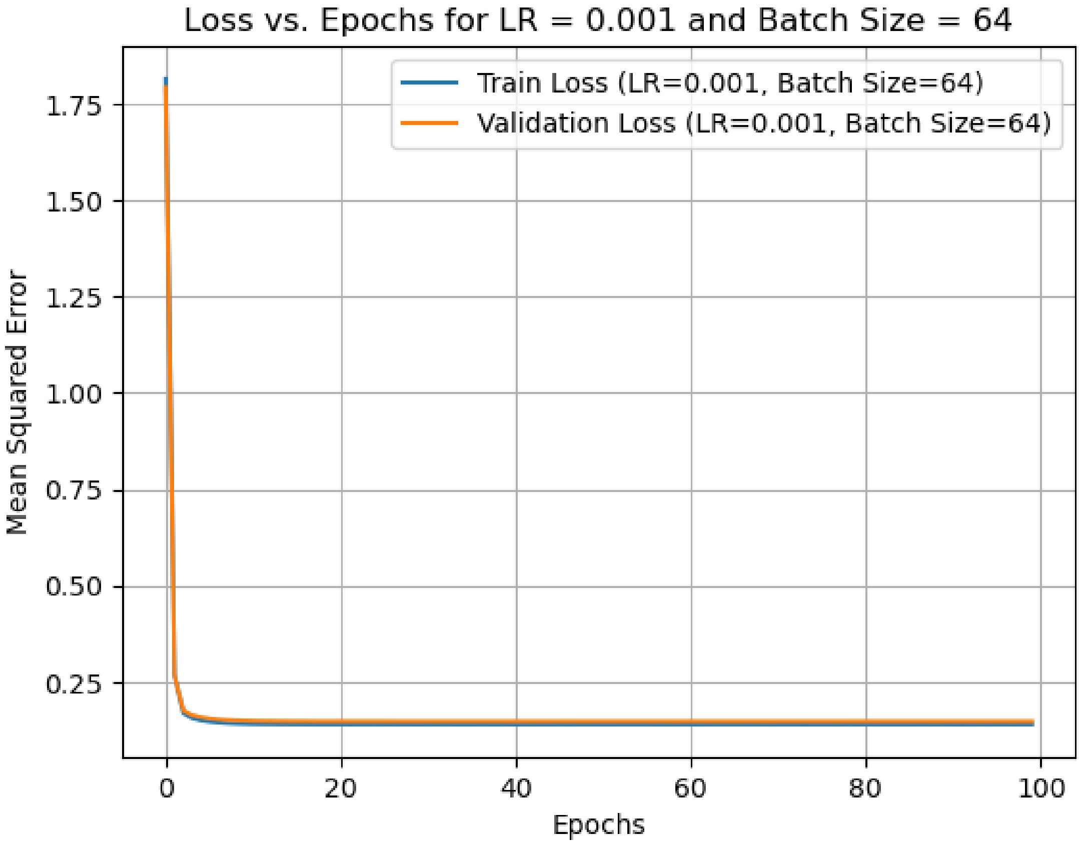
```
Training with Learning Rate = 0.001 and Batch Size = 16
Epoch 1: Train MSE = 1.8184, Validation MSE = 1.7956
Epoch 2: Train MSE = 0.2668, Validation MSE = 0.2673
Epoch 3: Train MSE = 0.1686, Validation MSE = 0.1754
Epoch 4: Train MSE = 0.1554, Validation MSE = 0.1636
Epoch 5: Train MSE = 0.1499, Validation MSE = 0.1583
Epoch 6: Train MSE = 0.1465, Validation MSE = 0.1548
Epoch 7: Train MSE = 0.1443, Validation MSE = 0.1525
Epoch 8: Train MSE = 0.1429, Validation MSE = 0.1509
Epoch 9: Train MSE = 0.1418, Validation MSE = 0.1498
Epoch 10: Train MSE = 0.1412, Validation MSE = 0.1490
Epoch 11: Train MSE = 0.1407, Validation MSE = 0.1485
Epoch 12: Train MSE = 0.1404, Validation MSE = 0.1481
Epoch 13: Train MSE = 0.1402, Validation MSE = 0.1478
Epoch 14: Train MSE = 0.1400, Validation MSE = 0.1476
Epoch 15: Train MSE = 0.1399, Validation MSE = 0.1474
Epoch 16: Train MSE = 0.1399, Validation MSE = 0.1473
Epoch 17: Train MSE = 0.1398, Validation MSE = 0.1472
Epoch 18: Train MSE = 0.1398, Validation MSE = 0.1471
Epoch 19: Train MSE = 0.1398, Validation MSE = 0.1471
Epoch 20: Train MSE = 0.1397, Validation MSE = 0.1471
Epoch 21: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 22: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 23: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 24: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 25: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 26: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 27: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 28: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 29: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 30: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 31: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 32: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 33: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 34: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 35: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 36: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 37: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 38: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 39: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 40: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 41: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 42: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 43: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 44: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 45: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 46: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 47: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 48: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 49: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 50: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 51: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 52: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 53: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 54: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 55: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 56: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 57: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 58: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 59: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 60: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 61: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 62: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 63: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 64: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 65: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 66: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 67: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 68: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 69: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 70: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 71: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 72: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 73: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 74: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 75: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 76: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 77: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 78: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 79: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 80: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 81: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 82: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 83: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 84: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 85: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 86: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 87: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 88: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 89: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 90: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 91: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 92: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 93: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 94: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 95: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 96: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 97: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 98: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 99: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 100: Train MSE = 0.1397, Validation MSE = 0.1468
```



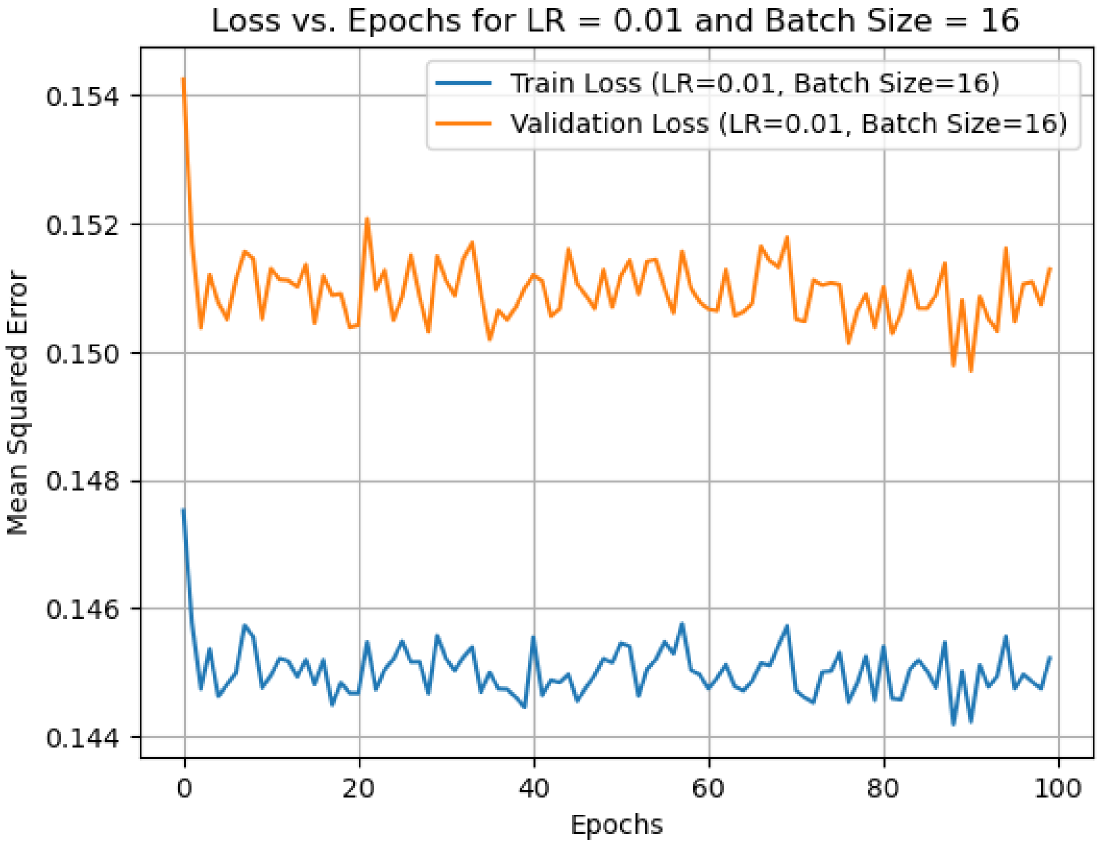
```
Training with Learning Rate = 0.001 and Batch Size = 32
Epoch 1: Train MSE = 1.8189, Validation MSE = 1.7959
Epoch 2: Train MSE = 0.2671, Validation MSE = 0.2675
Epoch 3: Train MSE = 0.1687, Validation MSE = 0.1756
Epoch 4: Train MSE = 0.1554, Validation MSE = 0.1637
Epoch 5: Train MSE = 0.1499, Validation MSE = 0.1584
Epoch 6: Train MSE = 0.1466, Validation MSE = 0.1549
Epoch 7: Train MSE = 0.1444, Validation MSE = 0.1525
Epoch 8: Train MSE = 0.1429, Validation MSE = 0.1509
Epoch 9: Train MSE = 0.1418, Validation MSE = 0.1498
Epoch 10: Train MSE = 0.1412, Validation MSE = 0.1490
Epoch 11: Train MSE = 0.1407, Validation MSE = 0.1484
Epoch 12: Train MSE = 0.1404, Validation MSE = 0.1480
Epoch 13: Train MSE = 0.1402, Validation MSE = 0.1478
Epoch 14: Train MSE = 0.1401, Validation MSE = 0.1476
Epoch 15: Train MSE = 0.1400, Validation MSE = 0.1474
Epoch 16: Train MSE = 0.1399, Validation MSE = 0.1473
Epoch 17: Train MSE = 0.1398, Validation MSE = 0.1472
Epoch 18: Train MSE = 0.1398, Validation MSE = 0.1472
Epoch 19: Train MSE = 0.1398, Validation MSE = 0.1471
Epoch 20: Train MSE = 0.1398, Validation MSE = 0.1470
Epoch 21: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 22: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 23: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 24: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 25: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 26: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 27: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 28: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 29: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 30: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 31: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 32: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 33: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 34: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 35: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 36: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 37: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 38: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 39: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 40: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 41: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 42: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 43: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 44: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 45: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 46: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 47: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 48: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 49: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 50: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 51: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 52: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 53: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 54: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 55: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 56: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 57: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 58: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 59: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 60: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 61: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 62: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 63: Train MSE = 0.1398, Validation MSE = 0.1468
Epoch 64: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 65: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 66: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 67: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 68: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 69: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 70: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 71: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 72: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 73: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 74: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 75: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 76: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 77: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 78: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 79: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 80: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 81: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 82: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 83: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 84: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 85: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 86: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 87: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 88: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 89: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 90: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 91: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 92: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 93: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 94: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 95: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 96: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 97: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 98: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 99: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 100: Train MSE = 0.1397, Validation MSE = 0.1468
```



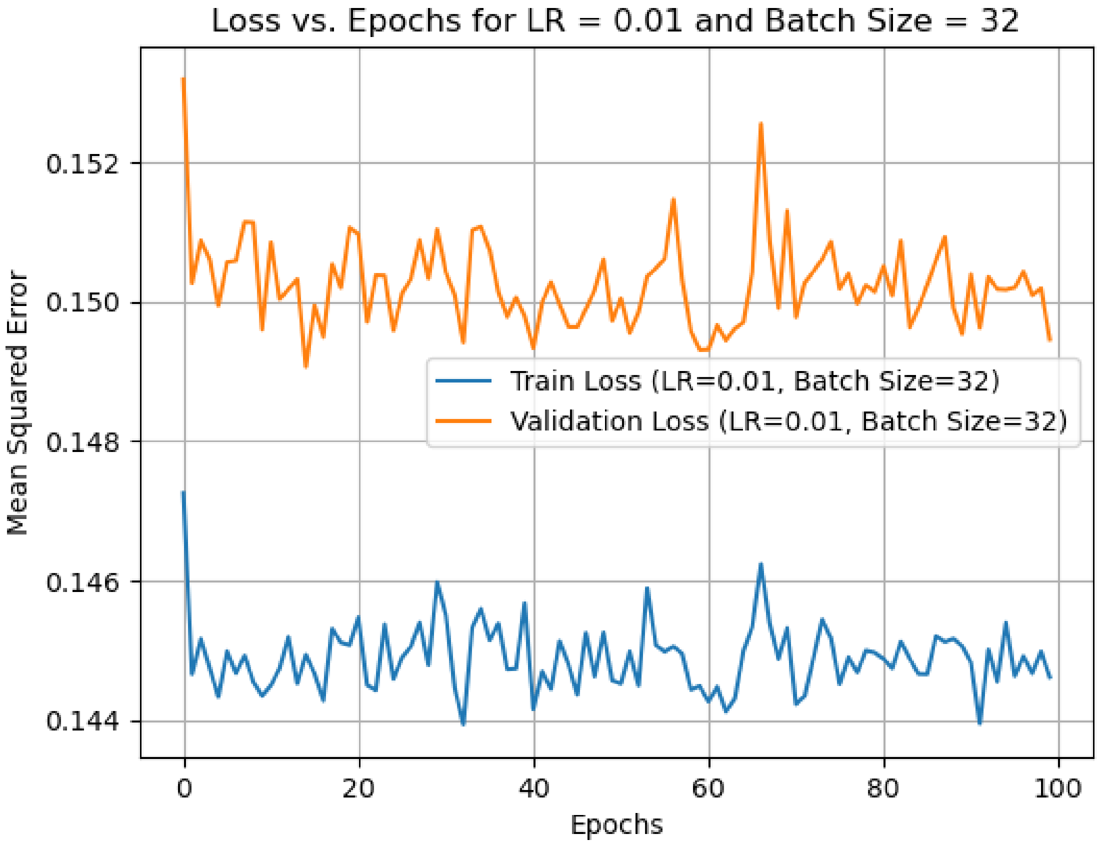
```
Training with Learning Rate = 0.001 and Batch Size = 64
Epoch 1: Train MSE = 1.8160, Validation MSE = 1.7933
Epoch 2: Train MSE = 0.2665, Validation MSE = 0.2671
Epoch 3: Train MSE = 0.1687, Validation MSE = 0.1755
Epoch 4: Train MSE = 0.1555, Validation MSE = 0.1638
Epoch 5: Train MSE = 0.1500, Validation MSE = 0.1584
Epoch 6: Train MSE = 0.1466, Validation MSE = 0.1549
Epoch 7: Train MSE = 0.1444, Validation MSE = 0.1525
Epoch 8: Train MSE = 0.1428, Validation MSE = 0.1509
Epoch 9: Train MSE = 0.1418, Validation MSE = 0.1497
Epoch 10: Train MSE = 0.1412, Validation MSE = 0.1489
Epoch 11: Train MSE = 0.1406, Validation MSE = 0.1484
Epoch 12: Train MSE = 0.1404, Validation MSE = 0.1480
Epoch 13: Train MSE = 0.1402, Validation MSE = 0.1477
Epoch 14: Train MSE = 0.1400, Validation MSE = 0.1475
Epoch 15: Train MSE = 0.1399, Validation MSE = 0.1474
Epoch 16: Train MSE = 0.1399, Validation MSE = 0.1472
Epoch 17: Train MSE = 0.1398, Validation MSE = 0.1472
Epoch 18: Train MSE = 0.1398, Validation MSE = 0.1471
Epoch 19: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 20: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 21: Train MSE = 0.1397, Validation MSE = 0.1470
Epoch 22: Train MSE = 0.1396, Validation MSE = 0.1469
Epoch 23: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 24: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 25: Train MSE = 0.1398, Validation MSE = 0.1469
Epoch 26: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 27: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 28: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 29: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 30: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 31: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 32: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 33: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 34: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 35: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 36: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 37: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 38: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 39: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 40: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 41: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 42: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 43: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 44: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 45: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 46: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 47: Train MSE = 0.1396, Validation MSE = 0.1469
Epoch 48: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 49: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 50: Train MSE = 0.1398, Validation MSE = 0.1468
Epoch 51: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 52: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 53: Train MSE = 0.1398, Validation MSE = 0.1468
Epoch 54: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 55: Train MSE = 0.1398, Validation MSE = 0.1468
Epoch 56: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 57: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 58: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 59: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 60: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 61: Train MSE = 0.1396, Validation MSE = 0.1469
Epoch 62: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 63: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 64: Train MSE = 0.1398, Validation MSE = 0.1468
Epoch 65: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 66: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 67: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 68: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 69: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 70: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 71: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 72: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 73: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 74: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 75: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 76: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 77: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 78: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 79: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 80: Train MSE = 0.1397, Validation MSE = 0.1469
Epoch 81: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 82: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 83: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 84: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 85: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 86: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 87: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 88: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 89: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 90: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 91: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 92: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 93: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 94: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 95: Train MSE = 0.1396, Validation MSE = 0.1468
Epoch 96: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 97: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 98: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 99: Train MSE = 0.1397, Validation MSE = 0.1468
Epoch 100: Train MSE = 0.1397, Validation MSE = 0.1468
```



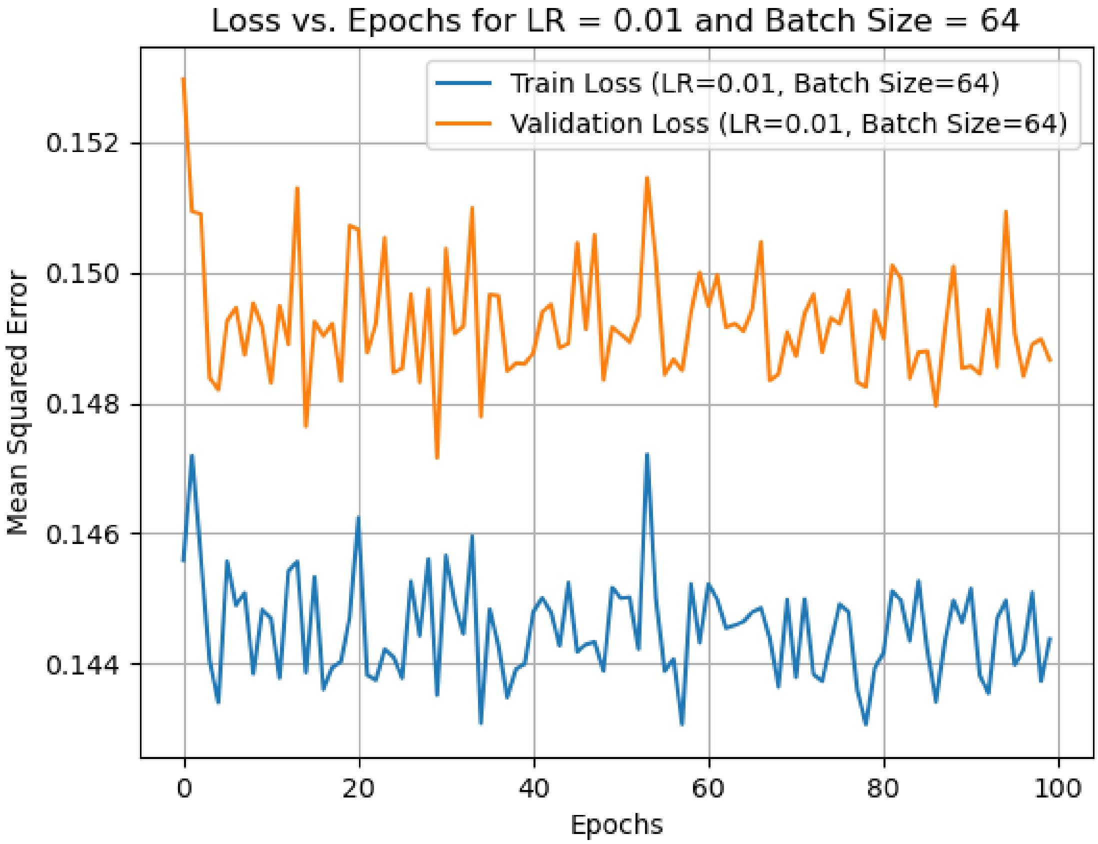
```
Training with Learning Rate = 0.01 and Batch Size = 16
Epoch 1: Train MSE = 0.1475, Validation MSE = 0.1542
Epoch 2: Train MSE = 0.1458, Validation MSE = 0.1517
Epoch 3: Train MSE = 0.1447, Validation MSE = 0.1504
Epoch 4: Train MSE = 0.1454, Validation MSE = 0.1512
Epoch 5: Train MSE = 0.1446, Validation MSE = 0.1508
Epoch 6: Train MSE = 0.1448, Validation MSE = 0.1505
Epoch 7: Train MSE = 0.1450, Validation MSE = 0.1511
Epoch 8: Train MSE = 0.1457, Validation MSE = 0.1516
Epoch 9: Train MSE = 0.1456, Validation MSE = 0.1515
Epoch 10: Train MSE = 0.1448, Validation MSE = 0.1505
Epoch 11: Train MSE = 0.1450, Validation MSE = 0.1513
Epoch 12: Train MSE = 0.1452, Validation MSE = 0.1511
Epoch 13: Train MSE = 0.1452, Validation MSE = 0.1511
Epoch 14: Train MSE = 0.1449, Validation MSE = 0.1510
Epoch 15: Train MSE = 0.1452, Validation MSE = 0.1514
Epoch 16: Train MSE = 0.1448, Validation MSE = 0.1504
Epoch 17: Train MSE = 0.1452, Validation MSE = 0.1512
Epoch 18: Train MSE = 0.1445, Validation MSE = 0.1509
Epoch 19: Train MSE = 0.1448, Validation MSE = 0.1509
Epoch 20: Train MSE = 0.1447, Validation MSE = 0.1504
Epoch 21: Train MSE = 0.1447, Validation MSE = 0.1504
Epoch 22: Train MSE = 0.1455, Validation MSE = 0.1521
Epoch 23: Train MSE = 0.1447, Validation MSE = 0.1510
Epoch 24: Train MSE = 0.1450, Validation MSE = 0.1513
Epoch 25: Train MSE = 0.1452, Validation MSE = 0.1505
Epoch 26: Train MSE = 0.1455, Validation MSE = 0.1509
Epoch 27: Train MSE = 0.1452, Validation MSE = 0.1515
Epoch 28: Train MSE = 0.1452, Validation MSE = 0.1508
Epoch 29: Train MSE = 0.1447, Validation MSE = 0.1503
Epoch 30: Train MSE = 0.1456, Validation MSE = 0.1515
Epoch 31: Train MSE = 0.1452, Validation MSE = 0.1511
Epoch 32: Train MSE = 0.1450, Validation MSE = 0.1509
Epoch 33: Train MSE = 0.1452, Validation MSE = 0.1514
Epoch 34: Train MSE = 0.1454, Validation MSE = 0.1517
Epoch 35: Train MSE = 0.1447, Validation MSE = 0.1509
Epoch 36: Train MSE = 0.1450, Validation MSE = 0.1502
Epoch 37: Train MSE = 0.1447, Validation MSE = 0.1506
Epoch 38: Train MSE = 0.1447, Validation MSE = 0.1505
Epoch 39: Train MSE = 0.1446, Validation MSE = 0.1507
Epoch 40: Train MSE = 0.1445, Validation MSE = 0.1510
Epoch 41: Train MSE = 0.1455, Validation MSE = 0.1512
Epoch 42: Train MSE = 0.1446, Validation MSE = 0.1511
Epoch 43: Train MSE = 0.1449, Validation MSE = 0.1506
Epoch 44: Train MSE = 0.1448, Validation MSE = 0.1507
Epoch 45: Train MSE = 0.1450, Validation MSE = 0.1516
Epoch 46: Train MSE = 0.1446, Validation MSE = 0.1511
Epoch 47: Train MSE = 0.1448, Validation MSE = 0.1509
Epoch 48: Train MSE = 0.1450, Validation MSE = 0.1507
Epoch 49: Train MSE = 0.1452, Validation MSE = 0.1513
Epoch 50: Train MSE = 0.1452, Validation MSE = 0.1507
Epoch 51: Train MSE = 0.1455, Validation MSE = 0.1512
Epoch 52: Train MSE = 0.1454, Validation MSE = 0.1514
Epoch 53: Train MSE = 0.1446, Validation MSE = 0.1509
Epoch 54: Train MSE = 0.1451, Validation MSE = 0.1514
Epoch 55: Train MSE = 0.1452, Validation MSE = 0.1514
Epoch 56: Train MSE = 0.1455, Validation MSE = 0.1510
Epoch 57: Train MSE = 0.1453, Validation MSE = 0.1506
Epoch 58: Train MSE = 0.1458, Validation MSE = 0.1516
Epoch 59: Train MSE = 0.1450, Validation MSE = 0.1510
Epoch 60: Train MSE = 0.1450, Validation MSE = 0.1508
Epoch 61: Train MSE = 0.1447, Validation MSE = 0.1507
Epoch 62: Train MSE = 0.1449, Validation MSE = 0.1506
Epoch 63: Train MSE = 0.1451, Validation MSE = 0.1513
Epoch 64: Train MSE = 0.1448, Validation MSE = 0.1506
Epoch 65: Train MSE = 0.1447, Validation MSE = 0.1506
Epoch 66: Train MSE = 0.1449, Validation MSE = 0.1508
Epoch 67: Train MSE = 0.1451, Validation MSE = 0.1516
Epoch 68: Train MSE = 0.1451, Validation MSE = 0.1514
Epoch 69: Train MSE = 0.1454, Validation MSE = 0.1513
Epoch 70: Train MSE = 0.1457, Validation MSE = 0.1518
Epoch 71: Train MSE = 0.1447, Validation MSE = 0.1505
Epoch 72: Train MSE = 0.1446, Validation MSE = 0.1505
Epoch 73: Train MSE = 0.1445, Validation MSE = 0.1511
Epoch 74: Train MSE = 0.1450, Validation MSE = 0.1510
Epoch 75: Train MSE = 0.1450, Validation MSE = 0.1511
Epoch 76: Train MSE = 0.1453, Validation MSE = 0.1510
Epoch 77: Train MSE = 0.1445, Validation MSE = 0.1501
Epoch 78: Train MSE = 0.1448, Validation MSE = 0.1506
Epoch 79: Train MSE = 0.1453, Validation MSE = 0.1509
Epoch 80: Train MSE = 0.1446, Validation MSE = 0.1504
Epoch 81: Train MSE = 0.1454, Validation MSE = 0.1510
Epoch 82: Train MSE = 0.1446, Validation MSE = 0.1503
Epoch 83: Train MSE = 0.1446, Validation MSE = 0.1506
Epoch 84: Train MSE = 0.1450, Validation MSE = 0.1513
Epoch 85: Train MSE = 0.1452, Validation MSE = 0.1507
Epoch 86: Train MSE = 0.1450, Validation MSE = 0.1507
Epoch 87: Train MSE = 0.1448, Validation MSE = 0.1509
Epoch 88: Train MSE = 0.1455, Validation MSE = 0.1514
Epoch 89: Train MSE = 0.1442, Validation MSE = 0.1498
Epoch 90: Train MSE = 0.1450, Validation MSE = 0.1508
Epoch 91: Train MSE = 0.1442, Validation MSE = 0.1497
Epoch 92: Train MSE = 0.1451, Validation MSE = 0.1509
Epoch 93: Train MSE = 0.1448, Validation MSE = 0.1505
Epoch 94: Train MSE = 0.1449, Validation MSE = 0.1503
Epoch 95: Train MSE = 0.1456, Validation MSE = 0.1516
Epoch 96: Train MSE = 0.1447, Validation MSE = 0.1505
Epoch 97: Train MSE = 0.1450, Validation MSE = 0.1511
Epoch 98: Train MSE = 0.1449, Validation MSE = 0.1511
Epoch 99: Train MSE = 0.1447, Validation MSE = 0.1507
Epoch 100: Train MSE = 0.1452, Validation MSE = 0.1513
```



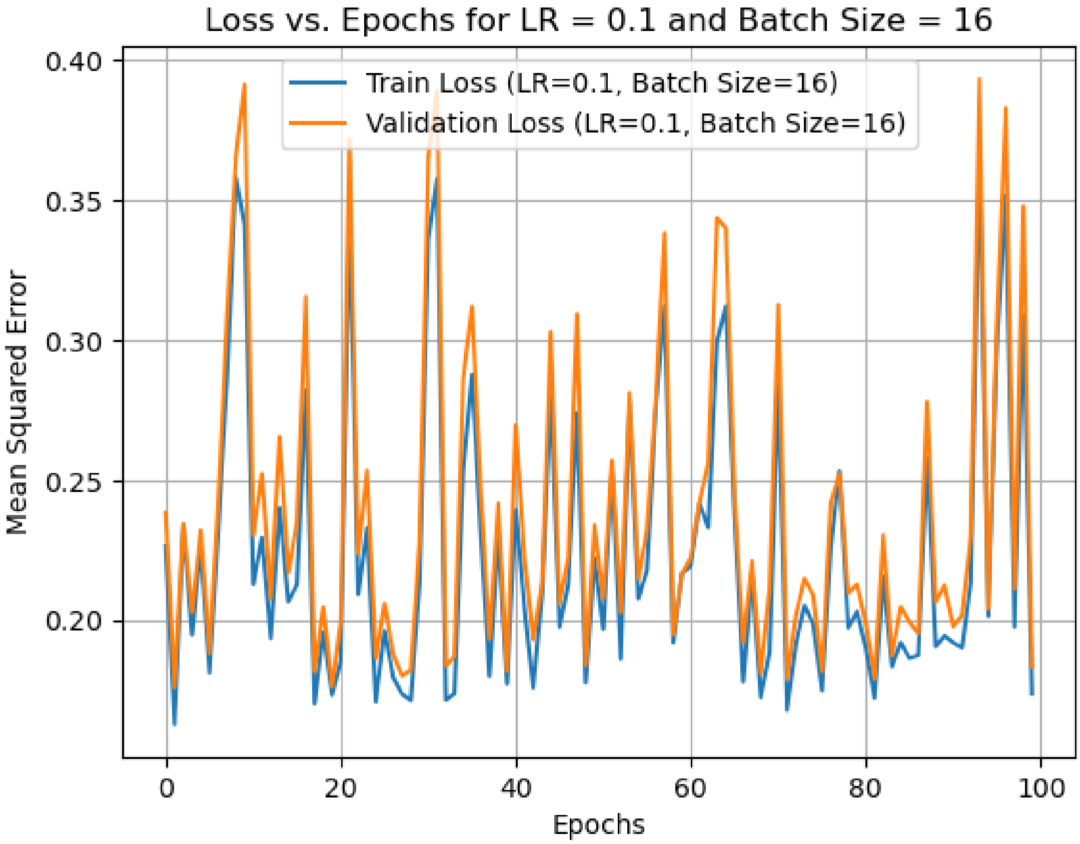
```
Training with Learning Rate = 0.01 and Batch Size = 32
Epoch 1: Train MSE = 0.1473, Validation MSE = 0.1532
Epoch 2: Train MSE = 0.1447, Validation MSE = 0.1503
Epoch 3: Train MSE = 0.1452, Validation MSE = 0.1509
Epoch 4: Train MSE = 0.1447, Validation MSE = 0.1506
Epoch 5: Train MSE = 0.1443, Validation MSE = 0.1499
Epoch 6: Train MSE = 0.1450, Validation MSE = 0.1506
Epoch 7: Train MSE = 0.1447, Validation MSE = 0.1506
Epoch 8: Train MSE = 0.1449, Validation MSE = 0.1511
Epoch 9: Train MSE = 0.1445, Validation MSE = 0.1511
Epoch 10: Train MSE = 0.1443, Validation MSE = 0.1496
Epoch 11: Train MSE = 0.1445, Validation MSE = 0.1508
Epoch 12: Train MSE = 0.1447, Validation MSE = 0.1500
Epoch 13: Train MSE = 0.1452, Validation MSE = 0.1502
Epoch 14: Train MSE = 0.1445, Validation MSE = 0.1503
Epoch 15: Train MSE = 0.1449, Validation MSE = 0.1491
Epoch 16: Train MSE = 0.1447, Validation MSE = 0.1499
Epoch 17: Train MSE = 0.1443, Validation MSE = 0.1495
Epoch 18: Train MSE = 0.1453, Validation MSE = 0.1505
Epoch 19: Train MSE = 0.1451, Validation MSE = 0.1502
Epoch 20: Train MSE = 0.1451, Validation MSE = 0.1511
Epoch 21: Train MSE = 0.1455, Validation MSE = 0.1510
Epoch 22: Train MSE = 0.1445, Validation MSE = 0.1497
Epoch 23: Train MSE = 0.1444, Validation MSE = 0.1504
Epoch 24: Train MSE = 0.1454, Validation MSE = 0.1504
Epoch 25: Train MSE = 0.1446, Validation MSE = 0.1496
Epoch 26: Train MSE = 0.1449, Validation MSE = 0.1501
Epoch 27: Train MSE = 0.1451, Validation MSE = 0.1503
Epoch 28: Train MSE = 0.1454, Validation MSE = 0.1509
Epoch 29: Train MSE = 0.1448, Validation MSE = 0.1503
Epoch 30: Train MSE = 0.1460, Validation MSE = 0.1510
Epoch 31: Train MSE = 0.1455, Validation MSE = 0.1504
Epoch 32: Train MSE = 0.1445, Validation MSE = 0.1501
Epoch 33: Train MSE = 0.1439, Validation MSE = 0.1494
Epoch 34: Train MSE = 0.1453, Validation MSE = 0.1510
Epoch 35: Train MSE = 0.1456, Validation MSE = 0.1511
Epoch 36: Train MSE = 0.1451, Validation MSE = 0.1507
Epoch 37: Train MSE = 0.1454, Validation MSE = 0.1501
Epoch 38: Train MSE = 0.1447, Validation MSE = 0.1498
Epoch 39: Train MSE = 0.1447, Validation MSE = 0.1501
Epoch 40: Train MSE = 0.1457, Validation MSE = 0.1498
Epoch 41: Train MSE = 0.1442, Validation MSE = 0.1493
Epoch 42: Train MSE = 0.1447, Validation MSE = 0.1500
Epoch 43: Train MSE = 0.1444, Validation MSE = 0.1503
Epoch 44: Train MSE = 0.1451, Validation MSE = 0.1500
Epoch 45: Train MSE = 0.1448, Validation MSE = 0.1496
Epoch 46: Train MSE = 0.1444, Validation MSE = 0.1496
Epoch 47: Train MSE = 0.1452, Validation MSE = 0.1499
Epoch 48: Train MSE = 0.1446, Validation MSE = 0.1502
Epoch 49: Train MSE = 0.1453, Validation MSE = 0.1506
Epoch 50: Train MSE = 0.1446, Validation MSE = 0.1497
Epoch 51: Train MSE = 0.1445, Validation MSE = 0.1500
Epoch 52: Train MSE = 0.1450, Validation MSE = 0.1495
Epoch 53: Train MSE = 0.1445, Validation MSE = 0.1498
Epoch 54: Train MSE = 0.1459, Validation MSE = 0.1504
Epoch 55: Train MSE = 0.1451, Validation MSE = 0.1505
Epoch 56: Train MSE = 0.1450, Validation MSE = 0.1506
Epoch 57: Train MSE = 0.1451, Validation MSE = 0.1515
Epoch 58: Train MSE = 0.1450, Validation MSE = 0.1503
Epoch 59: Train MSE = 0.1444, Validation MSE = 0.1496
Epoch 60: Train MSE = 0.1445, Validation MSE = 0.1493
Epoch 61: Train MSE = 0.1443, Validation MSE = 0.1493
Epoch 62: Train MSE = 0.1445, Validation MSE = 0.1497
Epoch 63: Train MSE = 0.1441, Validation MSE = 0.1494
Epoch 64: Train MSE = 0.1443, Validation MSE = 0.1496
Epoch 65: Train MSE = 0.1450, Validation MSE = 0.1497
Epoch 66: Train MSE = 0.1453, Validation MSE = 0.1504
Epoch 67: Train MSE = 0.1462, Validation MSE = 0.1525
Epoch 68: Train MSE = 0.1454, Validation MSE = 0.1509
Epoch 69: Train MSE = 0.1449, Validation MSE = 0.1499
Epoch 70: Train MSE = 0.1453, Validation MSE = 0.1513
Epoch 71: Train MSE = 0.1442, Validation MSE = 0.1498
Epoch 72: Train MSE = 0.1443, Validation MSE = 0.1503
Epoch 73: Train MSE = 0.1449, Validation MSE = 0.1504
Epoch 74: Train MSE = 0.1454, Validation MSE = 0.1506
Epoch 75: Train MSE = 0.1452, Validation MSE = 0.1508
Epoch 76: Train MSE = 0.1445, Validation MSE = 0.1502
Epoch 77: Train MSE = 0.1449, Validation MSE = 0.1504
Epoch 78: Train MSE = 0.1447, Validation MSE = 0.1500
Epoch 79: Train MSE = 0.1450, Validation MSE = 0.1502
Epoch 80: Train MSE = 0.1450, Validation MSE = 0.1501
Epoch 81: Train MSE = 0.1449, Validation MSE = 0.1505
Epoch 82: Train MSE = 0.1447, Validation MSE = 0.1501
Epoch 83: Train MSE = 0.1451, Validation MSE = 0.1509
Epoch 84: Train MSE = 0.1449, Validation MSE = 0.1496
Epoch 85: Train MSE = 0.1447, Validation MSE = 0.1499
Epoch 86: Train MSE = 0.1447, Validation MSE = 0.1502
Epoch 87: Train MSE = 0.1452, Validation MSE = 0.1506
Epoch 88: Train MSE = 0.1451, Validation MSE = 0.1509
Epoch 89: Train MSE = 0.1452, Validation MSE = 0.1499
Epoch 90: Train MSE = 0.1451, Validation MSE = 0.1495
Epoch 91: Train MSE = 0.1448, Validation MSE = 0.1504
Epoch 92: Train MSE = 0.1440, Validation MSE = 0.1496
Epoch 93: Train MSE = 0.1450, Validation MSE = 0.1503
Epoch 94: Train MSE = 0.1445, Validation MSE = 0.1502
Epoch 95: Train MSE = 0.1454, Validation MSE = 0.1502
Epoch 96: Train MSE = 0.1446, Validation MSE = 0.1502
Epoch 97: Train MSE = 0.1449, Validation MSE = 0.1504
Epoch 98: Train MSE = 0.1447, Validation MSE = 0.1501
Epoch 99: Train MSE = 0.1450, Validation MSE = 0.1502
Epoch 100: Train MSE = 0.1446, Validation MSE = 0.1495
```



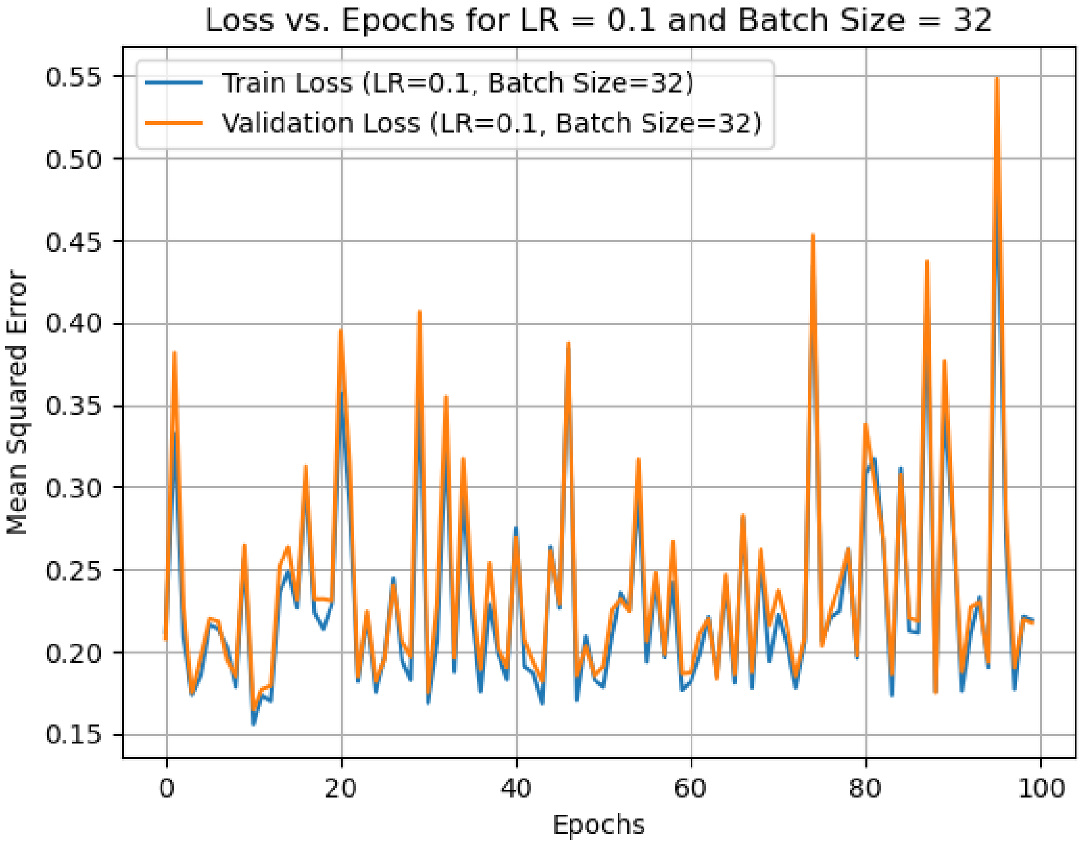
```
Training with Learning Rate = 0.01 and Batch Size = 64
Epoch 1: Train MSE = 0.1456, Validation MSE = 0.1530
Epoch 2: Train MSE = 0.1472, Validation MSE = 0.1509
Epoch 3: Train MSE = 0.1456, Validation MSE = 0.1509
Epoch 4: Train MSE = 0.1440, Validation MSE = 0.1484
Epoch 5: Train MSE = 0.1434, Validation MSE = 0.1482
Epoch 6: Train MSE = 0.1456, Validation MSE = 0.1493
Epoch 7: Train MSE = 0.1449, Validation MSE = 0.1495
Epoch 8: Train MSE = 0.1451, Validation MSE = 0.1487
Epoch 9: Train MSE = 0.1438, Validation MSE = 0.1495
Epoch 10: Train MSE = 0.1448, Validation MSE = 0.1492
Epoch 11: Train MSE = 0.1447, Validation MSE = 0.1483
Epoch 12: Train MSE = 0.1438, Validation MSE = 0.1495
Epoch 13: Train MSE = 0.1454, Validation MSE = 0.1489
Epoch 14: Train MSE = 0.1456, Validation MSE = 0.1513
Epoch 15: Train MSE = 0.1439, Validation MSE = 0.1476
Epoch 16: Train MSE = 0.1453, Validation MSE = 0.1492
Epoch 17: Train MSE = 0.1436, Validation MSE = 0.1490
Epoch 18: Train MSE = 0.1439, Validation MSE = 0.1492
Epoch 19: Train MSE = 0.1440, Validation MSE = 0.1483
Epoch 20: Train MSE = 0.1447, Validation MSE = 0.1507
Epoch 21: Train MSE = 0.1462, Validation MSE = 0.1507
Epoch 22: Train MSE = 0.1438, Validation MSE = 0.1488
Epoch 23: Train MSE = 0.1438, Validation MSE = 0.1492
Epoch 24: Train MSE = 0.1442, Validation MSE = 0.1505
Epoch 25: Train MSE = 0.1441, Validation MSE = 0.1485
Epoch 26: Train MSE = 0.1438, Validation MSE = 0.1485
Epoch 27: Train MSE = 0.1453, Validation MSE = 0.1497
Epoch 28: Train MSE = 0.1444, Validation MSE = 0.1483
Epoch 29: Train MSE = 0.1456, Validation MSE = 0.1497
Epoch 30: Train MSE = 0.1435, Validation MSE = 0.1472
Epoch 31: Train MSE = 0.1457, Validation MSE = 0.1504
Epoch 32: Train MSE = 0.1449, Validation MSE = 0.1491
Epoch 33: Train MSE = 0.1445, Validation MSE = 0.1492
Epoch 34: Train MSE = 0.1460, Validation MSE = 0.1510
Epoch 35: Train MSE = 0.1431, Validation MSE = 0.1478
Epoch 36: Train MSE = 0.1448, Validation MSE = 0.1497
Epoch 37: Train MSE = 0.1443, Validation MSE = 0.1496
Epoch 38: Train MSE = 0.1435, Validation MSE = 0.1485
Epoch 39: Train MSE = 0.1439, Validation MSE = 0.1486
Epoch 40: Train MSE = 0.1440, Validation MSE = 0.1486
Epoch 41: Train MSE = 0.1448, Validation MSE = 0.1488
Epoch 42: Train MSE = 0.1450, Validation MSE = 0.1494
Epoch 43: Train MSE = 0.1448, Validation MSE = 0.1495
Epoch 44: Train MSE = 0.1443, Validation MSE = 0.1488
Epoch 45: Train MSE = 0.1452, Validation MSE = 0.1489
Epoch 46: Train MSE = 0.1442, Validation MSE = 0.1505
Epoch 47: Train MSE = 0.1443, Validation MSE = 0.1491
Epoch 48: Train MSE = 0.1443, Validation MSE = 0.1506
Epoch 49: Train MSE = 0.1439, Validation MSE = 0.1484
Epoch 50: Train MSE = 0.1452, Validation MSE = 0.1492
Epoch 51: Train MSE = 0.1450, Validation MSE = 0.1491
Epoch 52: Train MSE = 0.1450, Validation MSE = 0.1489
Epoch 53: Train MSE = 0.1442, Validation MSE = 0.1493
Epoch 54: Train MSE = 0.1472, Validation MSE = 0.1515
Epoch 55: Train MSE = 0.1450, Validation MSE = 0.1502
Epoch 56: Train MSE = 0.1439, Validation MSE = 0.1484
Epoch 57: Train MSE = 0.1441, Validation MSE = 0.1487
Epoch 58: Train MSE = 0.1431, Validation MSE = 0.1485
Epoch 59: Train MSE = 0.1452, Validation MSE = 0.1494
Epoch 60: Train MSE = 0.1443, Validation MSE = 0.1500
Epoch 61: Train MSE = 0.1452, Validation MSE = 0.1495
Epoch 62: Train MSE = 0.1450, Validation MSE = 0.1500
Epoch 63: Train MSE = 0.1445, Validation MSE = 0.1492
Epoch 64: Train MSE = 0.1446, Validation MSE = 0.1492
Epoch 65: Train MSE = 0.1446, Validation MSE = 0.1491
Epoch 66: Train MSE = 0.1448, Validation MSE = 0.1494
Epoch 67: Train MSE = 0.1449, Validation MSE = 0.1505
Epoch 68: Train MSE = 0.1444, Validation MSE = 0.1483
Epoch 69: Train MSE = 0.1436, Validation MSE = 0.1484
Epoch 70: Train MSE = 0.1450, Validation MSE = 0.1491
Epoch 71: Train MSE = 0.1438, Validation MSE = 0.1487
Epoch 72: Train MSE = 0.1450, Validation MSE = 0.1494
Epoch 73: Train MSE = 0.1438, Validation MSE = 0.1497
Epoch 74: Train MSE = 0.1437, Validation MSE = 0.1488
Epoch 75: Train MSE = 0.1443, Validation MSE = 0.1493
Epoch 76: Train MSE = 0.1449, Validation MSE = 0.1492
Epoch 77: Train MSE = 0.1448, Validation MSE = 0.1497
Epoch 78: Train MSE = 0.1436, Validation MSE = 0.1483
Epoch 79: Train MSE = 0.1431, Validation MSE = 0.1482
Epoch 80: Train MSE = 0.1439, Validation MSE = 0.1494
Epoch 81: Train MSE = 0.1442, Validation MSE = 0.1490
Epoch 82: Train MSE = 0.1451, Validation MSE = 0.1501
Epoch 83: Train MSE = 0.1450, Validation MSE = 0.1499
Epoch 84: Train MSE = 0.1443, Validation MSE = 0.1484
Epoch 85: Train MSE = 0.1453, Validation MSE = 0.1488
Epoch 86: Train MSE = 0.1442, Validation MSE = 0.1488
Epoch 87: Train MSE = 0.1434, Validation MSE = 0.1480
Epoch 88: Train MSE = 0.1443, Validation MSE = 0.1491
Epoch 89: Train MSE = 0.1450, Validation MSE = 0.1501
Epoch 90: Train MSE = 0.1446, Validation MSE = 0.1485
Epoch 91: Train MSE = 0.1452, Validation MSE = 0.1486
Epoch 92: Train MSE = 0.1438, Validation MSE = 0.1484
Epoch 93: Train MSE = 0.1435, Validation MSE = 0.1494
Epoch 94: Train MSE = 0.1447, Validation MSE = 0.1486
Epoch 95: Train MSE = 0.1450, Validation MSE = 0.1509
Epoch 96: Train MSE = 0.1440, Validation MSE = 0.1491
Epoch 97: Train MSE = 0.1442, Validation MSE = 0.1484
Epoch 98: Train MSE = 0.1451, Validation MSE = 0.1489
Epoch 99: Train MSE = 0.1437, Validation MSE = 0.1490
Epoch 100: Train MSE = 0.1444, Validation MSE = 0.1487
```



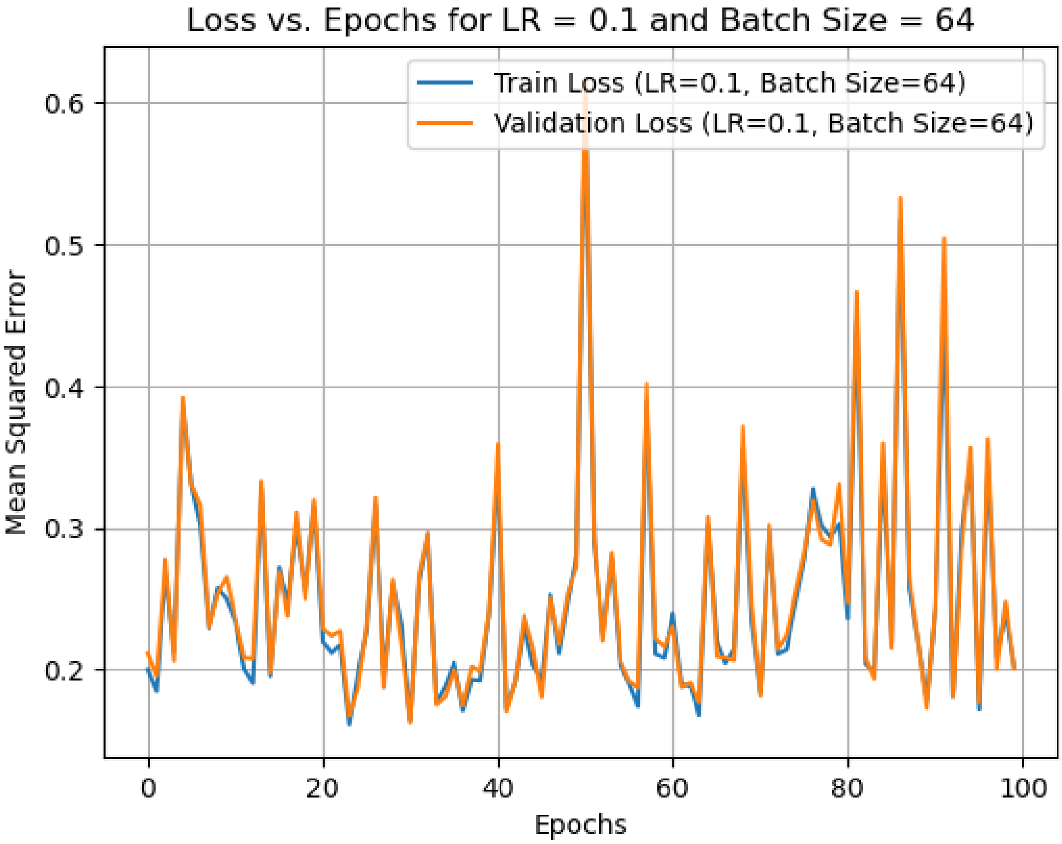
```
Training with Learning Rate = 0.1 and Batch Size = 16
Epoch 1: Train MSE = 0.2265, Validation MSE = 0.2383
Epoch 2: Train MSE = 0.1627, Validation MSE = 0.1759
Epoch 3: Train MSE = 0.2338, Validation MSE = 0.2344
Epoch 4: Train MSE = 0.1947, Validation MSE = 0.2028
Epoch 5: Train MSE = 0.2262, Validation MSE = 0.2321
Epoch 6: Train MSE = 0.1811, Validation MSE = 0.1881
Epoch 7: Train MSE = 0.2325, Validation MSE = 0.2425
Epoch 8: Train MSE = 0.2868, Validation MSE = 0.3084
Epoch 9: Train MSE = 0.3583, Validation MSE = 0.3656
Epoch 10: Train MSE = 0.3412, Validation MSE = 0.3914
Epoch 11: Train MSE = 0.2128, Validation MSE = 0.2302
Epoch 12: Train MSE = 0.2294, Validation MSE = 0.2523
Epoch 13: Train MSE = 0.1934, Validation MSE = 0.2077
Epoch 14: Train MSE = 0.2401, Validation MSE = 0.2655
Epoch 15: Train MSE = 0.2066, Validation MSE = 0.2170
Epoch 16: Train MSE = 0.2126, Validation MSE = 0.2352
Epoch 17: Train MSE = 0.2820, Validation MSE = 0.3156
Epoch 18: Train MSE = 0.1702, Validation MSE = 0.1817
Epoch 19: Train MSE = 0.1957, Validation MSE = 0.2046
Epoch 20: Train MSE = 0.1732, Validation MSE = 0.1761
Epoch 21: Train MSE = 0.1855, Validation MSE = 0.1996
Epoch 22: Train MSE = 0.3461, Validation MSE = 0.3718
Epoch 23: Train MSE = 0.2091, Validation MSE = 0.2238
Epoch 24: Train MSE = 0.2330, Validation MSE = 0.2536
Epoch 25: Train MSE = 0.1709, Validation MSE = 0.1863
Epoch 26: Train MSE = 0.1961, Validation MSE = 0.2058
Epoch 27: Train MSE = 0.1795, Validation MSE = 0.1877
Epoch 28: Train MSE = 0.1735, Validation MSE = 0.1801
Epoch 29: Train MSE = 0.1713, Validation MSE = 0.1820
Epoch 30: Train MSE = 0.2131, Validation MSE = 0.2274
Epoch 31: Train MSE = 0.3361, Validation MSE = 0.3659
Epoch 32: Train MSE = 0.3577, Validation MSE = 0.3892
Epoch 33: Train MSE = 0.1715, Validation MSE = 0.1837
Epoch 34: Train MSE = 0.1737, Validation MSE = 0.1870
Epoch 35: Train MSE = 0.2537, Validation MSE = 0.2865
Epoch 36: Train MSE = 0.2878, Validation MSE = 0.3120
Epoch 37: Train MSE = 0.2307, Validation MSE = 0.2445
Epoch 38: Train MSE = 0.1800, Validation MSE = 0.1933
Epoch 39: Train MSE = 0.2307, Validation MSE = 0.2418
Epoch 40: Train MSE = 0.1772, Validation MSE = 0.1818
Epoch 41: Train MSE = 0.2394, Validation MSE = 0.2697
Epoch 42: Train MSE = 0.2040, Validation MSE = 0.2217
Epoch 43: Train MSE = 0.1758, Validation MSE = 0.1931
Epoch 44: Train MSE = 0.2090, Validation MSE = 0.2129
Epoch 45: Train MSE = 0.2808, Validation MSE = 0.3030
Epoch 46: Train MSE = 0.1975, Validation MSE = 0.2059
Epoch 47: Train MSE = 0.2128, Validation MSE = 0.2223
Epoch 48: Train MSE = 0.2741, Validation MSE = 0.3094
Epoch 49: Train MSE = 0.1776, Validation MSE = 0.1839
Epoch 50: Train MSE = 0.2222, Validation MSE = 0.2340
Epoch 51: Train MSE = 0.1968, Validation MSE = 0.2078
Epoch 52: Train MSE = 0.2530, Validation MSE = 0.2571
Epoch 53: Train MSE = 0.1861, Validation MSE = 0.2027
Epoch 54: Train MSE = 0.2778, Validation MSE = 0.2812
Epoch 55: Train MSE = 0.2076, Validation MSE = 0.2149
Epoch 56: Train MSE = 0.2177, Validation MSE = 0.2323
Epoch 57: Train MSE = 0.2761, Validation MSE = 0.2786
Epoch 58: Train MSE = 0.3123, Validation MSE = 0.3382
Epoch 59: Train MSE = 0.1920, Validation MSE = 0.1947
Epoch 60: Train MSE = 0.2166, Validation MSE = 0.2165
Epoch 61: Train MSE = 0.2194, Validation MSE = 0.2224
Epoch 62: Train MSE = 0.2416, Validation MSE = 0.2421
Epoch 63: Train MSE = 0.2330, Validation MSE = 0.2560
Epoch 64: Train MSE = 0.2994, Validation MSE = 0.3437
Epoch 65: Train MSE = 0.3121, Validation MSE = 0.3401
Epoch 66: Train MSE = 0.2373, Validation MSE = 0.2431
Epoch 67: Train MSE = 0.1779, Validation MSE = 0.1920
Epoch 68: Train MSE = 0.2162, Validation MSE = 0.2210
Epoch 69: Train MSE = 0.1724, Validation MSE = 0.1810
Epoch 70: Train MSE = 0.1884, Validation MSE = 0.2109
Epoch 71: Train MSE = 0.2842, Validation MSE = 0.3126
Epoch 72: Train MSE = 0.1678, Validation MSE = 0.1788
Epoch 73: Train MSE = 0.1912, Validation MSE = 0.2015
Epoch 74: Train MSE = 0.2052, Validation MSE = 0.2148
Epoch 75: Train MSE = 0.1986, Validation MSE = 0.2089
Epoch 76: Train MSE = 0.1748, Validation MSE = 0.1816
Epoch 77: Train MSE = 0.2264, Validation MSE = 0.2418
Epoch 78: Train MSE = 0.2532, Validation MSE = 0.2525
Epoch 79: Train MSE = 0.1972, Validation MSE = 0.2096
Epoch 80: Train MSE = 0.2030, Validation MSE = 0.2127
Epoch 81: Train MSE = 0.1901, Validation MSE = 0.1990
Epoch 82: Train MSE = 0.1721, Validation MSE = 0.1790
Epoch 83: Train MSE = 0.2155, Validation MSE = 0.2304
Epoch 84: Train MSE = 0.1835, Validation MSE = 0.1873
Epoch 85: Train MSE = 0.1919, Validation MSE = 0.2047
Epoch 86: Train MSE = 0.1864, Validation MSE = 0.1994
Epoch 87: Train MSE = 0.1874, Validation MSE = 0.1951
Epoch 88: Train MSE = 0.2581, Validation MSE = 0.2779
Epoch 89: Train MSE = 0.1906, Validation MSE = 0.2065
Epoch 90: Train MSE = 0.1943, Validation MSE = 0.2125
Epoch 91: Train MSE = 0.1919, Validation MSE = 0.1977
Epoch 92: Train MSE = 0.1902, Validation MSE = 0.2017
Epoch 93: Train MSE = 0.2132, Validation MSE = 0.2303
Epoch 94: Train MSE = 0.3690, Validation MSE = 0.3935
Epoch 95: Train MSE = 0.2014, Validation MSE = 0.2040
Epoch 96: Train MSE = 0.2992, Validation MSE = 0.3063
Epoch 97: Train MSE = 0.3518, Validation MSE = 0.3830
Epoch 98: Train MSE = 0.1976, Validation MSE = 0.2112
Epoch 99: Train MSE = 0.3091, Validation MSE = 0.3479
Epoch 100: Train MSE = 0.1736, Validation MSE = 0.1834
```



```
Training with Learning Rate = 0.1 and Batch Size = 32
Epoch 1: Train MSE = 0.2119, Validation MSE = 0.2077
Epoch 2: Train MSE = 0.3325, Validation MSE = 0.3815
Epoch 3: Train MSE = 0.2085, Validation MSE = 0.2270
Epoch 4: Train MSE = 0.1735, Validation MSE = 0.1748
Epoch 5: Train MSE = 0.1855, Validation MSE = 0.1967
Epoch 6: Train MSE = 0.2162, Validation MSE = 0.2200
Epoch 7: Train MSE = 0.2138, Validation MSE = 0.2182
Epoch 8: Train MSE = 0.2023, Validation MSE = 0.1948
Epoch 9: Train MSE = 0.1784, Validation MSE = 0.1847
Epoch 10: Train MSE = 0.2570, Validation MSE = 0.2645
Epoch 11: Train MSE = 0.1556, Validation MSE = 0.1646
Epoch 12: Train MSE = 0.1729, Validation MSE = 0.1769
Epoch 13: Train MSE = 0.1698, Validation MSE = 0.1791
Epoch 14: Train MSE = 0.2357, Validation MSE = 0.2520
Epoch 15: Train MSE = 0.2484, Validation MSE = 0.2635
Epoch 16: Train MSE = 0.2265, Validation MSE = 0.2312
Epoch 17: Train MSE = 0.3032, Validation MSE = 0.3125
Epoch 18: Train MSE = 0.2235, Validation MSE = 0.2316
Epoch 19: Train MSE = 0.2135, Validation MSE = 0.2317
Epoch 20: Train MSE = 0.2290, Validation MSE = 0.2307
Epoch 21: Train MSE = 0.3569, Validation MSE = 0.3949
Epoch 22: Train MSE = 0.2886, Validation MSE = 0.3129
Epoch 23: Train MSE = 0.1818, Validation MSE = 0.1847
Epoch 24: Train MSE = 0.2212, Validation MSE = 0.2242
Epoch 25: Train MSE = 0.1750, Validation MSE = 0.1817
Epoch 26: Train MSE = 0.1963, Validation MSE = 0.1950
Epoch 27: Train MSE = 0.2444, Validation MSE = 0.2401
Epoch 28: Train MSE = 0.1945, Validation MSE = 0.2063
Epoch 29: Train MSE = 0.1827, Validation MSE = 0.1967
Epoch 30: Train MSE = 0.3588, Validation MSE = 0.4065
Epoch 31: Train MSE = 0.1688, Validation MSE = 0.1749
Epoch 32: Train MSE = 0.2058, Validation MSE = 0.2289
Epoch 33: Train MSE = 0.3278, Validation MSE = 0.3547
Epoch 34: Train MSE = 0.1875, Validation MSE = 0.1960
Epoch 35: Train MSE = 0.2942, Validation MSE = 0.3171
Epoch 36: Train MSE = 0.2206, Validation MSE = 0.2349
Epoch 37: Train MSE = 0.1754, Validation MSE = 0.1890
Epoch 38: Train MSE = 0.2284, Validation MSE = 0.2537
Epoch 39: Train MSE = 0.1978, Validation MSE = 0.2018
Epoch 40: Train MSE = 0.1830, Validation MSE = 0.1902
Epoch 41: Train MSE = 0.2748, Validation MSE = 0.2691
Epoch 42: Train MSE = 0.1910, Validation MSE = 0.2073
Epoch 43: Train MSE = 0.1865, Validation MSE = 0.1933
Epoch 44: Train MSE = 0.1684, Validation MSE = 0.1818
Epoch 45: Train MSE = 0.2636, Validation MSE = 0.2610
Epoch 46: Train MSE = 0.2263, Validation MSE = 0.2290
Epoch 47: Train MSE = 0.3840, Validation MSE = 0.3872
Epoch 48: Train MSE = 0.1703, Validation MSE = 0.1853
Epoch 49: Train MSE = 0.2090, Validation MSE = 0.2031
Epoch 50: Train MSE = 0.1828, Validation MSE = 0.1850
Epoch 51: Train MSE = 0.1783, Validation MSE = 0.1909
Epoch 52: Train MSE = 0.2107, Validation MSE = 0.2255
Epoch 53: Train MSE = 0.2356, Validation MSE = 0.2318
Epoch 54: Train MSE = 0.2250, Validation MSE = 0.2243
Epoch 55: Train MSE = 0.2950, Validation MSE = 0.3168
Epoch 56: Train MSE = 0.1936, Validation MSE = 0.2062
Epoch 57: Train MSE = 0.2428, Validation MSE = 0.2478
Epoch 58: Train MSE = 0.1964, Validation MSE = 0.1983
Epoch 59: Train MSE = 0.2421, Validation MSE = 0.2668
Epoch 60: Train MSE = 0.1764, Validation MSE = 0.1864
Epoch 61: Train MSE = 0.1817, Validation MSE = 0.1871
Epoch 62: Train MSE = 0.1976, Validation MSE = 0.2106
Epoch 63: Train MSE = 0.2209, Validation MSE = 0.2197
Epoch 64: Train MSE = 0.1850, Validation MSE = 0.1836
Epoch 65: Train MSE = 0.2408, Validation MSE = 0.2465
Epoch 66: Train MSE = 0.1809, Validation MSE = 0.1860
Epoch 67: Train MSE = 0.2807, Validation MSE = 0.2826
Epoch 68: Train MSE = 0.1778, Validation MSE = 0.1874
Epoch 69: Train MSE = 0.2555, Validation MSE = 0.2618
Epoch 70: Train MSE = 0.1939, Validation MSE = 0.2157
Epoch 71: Train MSE = 0.2221, Validation MSE = 0.2370
Epoch 72: Train MSE = 0.2040, Validation MSE = 0.2153
Epoch 73: Train MSE = 0.1778, Validation MSE = 0.1849
Epoch 74: Train MSE = 0.2059, Validation MSE = 0.2083
Epoch 75: Train MSE = 0.4345, Validation MSE = 0.4531
Epoch 76: Train MSE = 0.2065, Validation MSE = 0.2033
Epoch 77: Train MSE = 0.2210, Validation MSE = 0.2255
Epoch 78: Train MSE = 0.2243, Validation MSE = 0.2411
Epoch 79: Train MSE = 0.2623, Validation MSE = 0.2619
Epoch 80: Train MSE = 0.1960, Validation MSE = 0.1976
Epoch 81: Train MSE = 0.3081, Validation MSE = 0.3381
Epoch 82: Train MSE = 0.3171, Validation MSE = 0.3029
Epoch 83: Train MSE = 0.2661, Validation MSE = 0.2684
Epoch 84: Train MSE = 0.1732, Validation MSE = 0.1861
Epoch 85: Train MSE = 0.3112, Validation MSE = 0.3074
Epoch 86: Train MSE = 0.2124, Validation MSE = 0.2204
Epoch 87: Train MSE = 0.2112, Validation MSE = 0.2184
Epoch 88: Train MSE = 0.4091, Validation MSE = 0.4372
Epoch 89: Train MSE = 0.1753, Validation MSE = 0.1749
Epoch 90: Train MSE = 0.3482, Validation MSE = 0.3766
Epoch 91: Train MSE = 0.2754, Validation MSE = 0.2780
Epoch 92: Train MSE = 0.1758, Validation MSE = 0.1876
Epoch 93: Train MSE = 0.2112, Validation MSE = 0.2270
Epoch 94: Train MSE = 0.2329, Validation MSE = 0.2299
Epoch 95: Train MSE = 0.1903, Validation MSE = 0.1934
Epoch 96: Train MSE = 0.4950, Validation MSE = 0.5482
Epoch 97: Train MSE = 0.2696, Validation MSE = 0.2864
Epoch 98: Train MSE = 0.1770, Validation MSE = 0.1901
Epoch 99: Train MSE = 0.2212, Validation MSE = 0.2195
Epoch 100: Train MSE = 0.2192, Validation MSE = 0.2173
```



```
Training with Learning Rate = 0.1 and Batch Size = 64
Epoch 1: Train MSE = 0.1998, Validation MSE = 0.2115
Epoch 2: Train MSE = 0.1846, Validation MSE = 0.1952
Epoch 3: Train MSE = 0.2679, Validation MSE = 0.2774
Epoch 4: Train MSE = 0.2126, Validation MSE = 0.2061
Epoch 5: Train MSE = 0.3869, Validation MSE = 0.3918
Epoch 6: Train MSE = 0.3299, Validation MSE = 0.3300
Epoch 7: Train MSE = 0.3012, Validation MSE = 0.3160
Epoch 8: Train MSE = 0.2291, Validation MSE = 0.2299
Epoch 9: Train MSE = 0.2573, Validation MSE = 0.2536
Epoch 10: Train MSE = 0.2499, Validation MSE = 0.2651
Epoch 11: Train MSE = 0.2335, Validation MSE = 0.2364
Epoch 12: Train MSE = 0.2002, Validation MSE = 0.2084
Epoch 13: Train MSE = 0.1902, Validation MSE = 0.2076
Epoch 14: Train MSE = 0.3271, Validation MSE = 0.3327
Epoch 15: Train MSE = 0.1954, Validation MSE = 0.1977
Epoch 16: Train MSE = 0.2720, Validation MSE = 0.2690
Epoch 17: Train MSE = 0.2472, Validation MSE = 0.2378
Epoch 18: Train MSE = 0.3005, Validation MSE = 0.3106
Epoch 19: Train MSE = 0.2544, Validation MSE = 0.2498
Epoch 20: Train MSE = 0.3146, Validation MSE = 0.3194
Epoch 21: Train MSE = 0.2194, Validation MSE = 0.2286
Epoch 22: Train MSE = 0.2115, Validation MSE = 0.2234
Epoch 23: Train MSE = 0.2173, Validation MSE = 0.2271
Epoch 24: Train MSE = 0.1612, Validation MSE = 0.1673
Epoch 25: Train MSE = 0.1971, Validation MSE = 0.1863
Epoch 26: Train MSE = 0.2263, Validation MSE = 0.2312
Epoch 27: Train MSE = 0.3158, Validation MSE = 0.3214
Epoch 28: Train MSE = 0.1952, Validation MSE = 0.1872
Epoch 29: Train MSE = 0.2602, Validation MSE = 0.2634
Epoch 30: Train MSE = 0.2304, Validation MSE = 0.2155
Epoch 31: Train MSE = 0.1639, Validation MSE = 0.1624
Epoch 32: Train MSE = 0.2624, Validation MSE = 0.2669
Epoch 33: Train MSE = 0.2952, Validation MSE = 0.2968
Epoch 34: Train MSE = 0.1766, Validation MSE = 0.1754
Epoch 35: Train MSE = 0.1881, Validation MSE = 0.1806
Epoch 36: Train MSE = 0.2049, Validation MSE = 0.1997
Epoch 37: Train MSE = 0.1710, Validation MSE = 0.1746
Epoch 38: Train MSE = 0.1927, Validation MSE = 0.2020
Epoch 39: Train MSE = 0.1920, Validation MSE = 0.1983
Epoch 40: Train MSE = 0.2400, Validation MSE = 0.2404
Epoch 41: Train MSE = 0.3415, Validation MSE = 0.3590
Epoch 42: Train MSE = 0.1726, Validation MSE = 0.1704
Epoch 43: Train MSE = 0.1917, Validation MSE = 0.1924
Epoch 44: Train MSE = 0.2307, Validation MSE = 0.2380
Epoch 45: Train MSE = 0.2032, Validation MSE = 0.2154
Epoch 46: Train MSE = 0.1923, Validation MSE = 0.1804
Epoch 47: Train MSE = 0.2528, Validation MSE = 0.2505
Epoch 48: Train MSE = 0.2115, Validation MSE = 0.2177
Epoch 49: Train MSE = 0.2490, Validation MSE = 0.2541
Epoch 50: Train MSE = 0.2813, Validation MSE = 0.2722
Epoch 51: Train MSE = 0.5932, Validation MSE = 0.6167
Epoch 52: Train MSE = 0.2884, Validation MSE = 0.2941
Epoch 53: Train MSE = 0.2249, Validation MSE = 0.2203
Epoch 54: Train MSE = 0.2773, Validation MSE = 0.2822
Epoch 55: Train MSE = 0.2028, Validation MSE = 0.2063
Epoch 56: Train MSE = 0.1903, Validation MSE = 0.1918
Epoch 57: Train MSE = 0.1739, Validation MSE = 0.1871
Epoch 58: Train MSE = 0.3893, Validation MSE = 0.4017
Epoch 59: Train MSE = 0.2111, Validation MSE = 0.2217
Epoch 60: Train MSE = 0.2081, Validation MSE = 0.2160
Epoch 61: Train MSE = 0.2393, Validation MSE = 0.2303
Epoch 62: Train MSE = 0.1894, Validation MSE = 0.1874
Epoch 63: Train MSE = 0.1883, Validation MSE = 0.1906
Epoch 64: Train MSE = 0.1677, Validation MSE = 0.1768
Epoch 65: Train MSE = 0.3011, Validation MSE = 0.3075
Epoch 66: Train MSE = 0.2206, Validation MSE = 0.2091
Epoch 67: Train MSE = 0.2044, Validation MSE = 0.2084
Epoch 68: Train MSE = 0.2146, Validation MSE = 0.2065
Epoch 69: Train MSE = 0.3556, Validation MSE = 0.3715
Epoch 70: Train MSE = 0.2334, Validation MSE = 0.2450
Epoch 71: Train MSE = 0.1838, Validation MSE = 0.1814
Epoch 72: Train MSE = 0.2980, Validation MSE = 0.3017
Epoch 73: Train MSE = 0.2110, Validation MSE = 0.2143
Epoch 74: Train MSE = 0.2140, Validation MSE = 0.2240
Epoch 75: Train MSE = 0.2465, Validation MSE = 0.2534
Epoch 76: Train MSE = 0.2777, Validation MSE = 0.2827
Epoch 77: Train MSE = 0.3273, Validation MSE = 0.3190
Epoch 78: Train MSE = 0.3014, Validation MSE = 0.2918
Epoch 79: Train MSE = 0.2935, Validation MSE = 0.2879
Epoch 80: Train MSE = 0.3026, Validation MSE = 0.3308
Epoch 81: Train MSE = 0.2359, Validation MSE = 0.2470
Epoch 82: Train MSE = 0.4459, Validation MSE = 0.4661
Epoch 83: Train MSE = 0.2043, Validation MSE = 0.2088
Epoch 84: Train MSE = 0.1981, Validation MSE = 0.1934
Epoch 85: Train MSE = 0.3424, Validation MSE = 0.3595
Epoch 86: Train MSE = 0.2191, Validation MSE = 0.2151
Epoch 87: Train MSE = 0.5167, Validation MSE = 0.5326
Epoch 88: Train MSE = 0.2571, Validation MSE = 0.2655
Epoch 89: Train MSE = 0.2202, Validation MSE = 0.2217
Epoch 90: Train MSE = 0.1782, Validation MSE = 0.1728
Epoch 91: Train MSE = 0.2427, Validation MSE = 0.2439
Epoch 92: Train MSE = 0.4523, Validation MSE = 0.5041
Epoch 93: Train MSE = 0.1807, Validation MSE = 0.1810
Epoch 94: Train MSE = 0.2995, Validation MSE = 0.2864
Epoch 95: Train MSE = 0.3454, Validation MSE = 0.3566
Epoch 96: Train MSE = 0.1718, Validation MSE = 0.1768
Epoch 97: Train MSE = 0.3460, Validation MSE = 0.3624
Epoch 98: Train MSE = 0.2088, Validation MSE = 0.2001
Epoch 99: Train MSE = 0.2395, Validation MSE = 0.2477
Epoch 100: Train MSE = 0.2013, Validation MSE = 0.2008
```



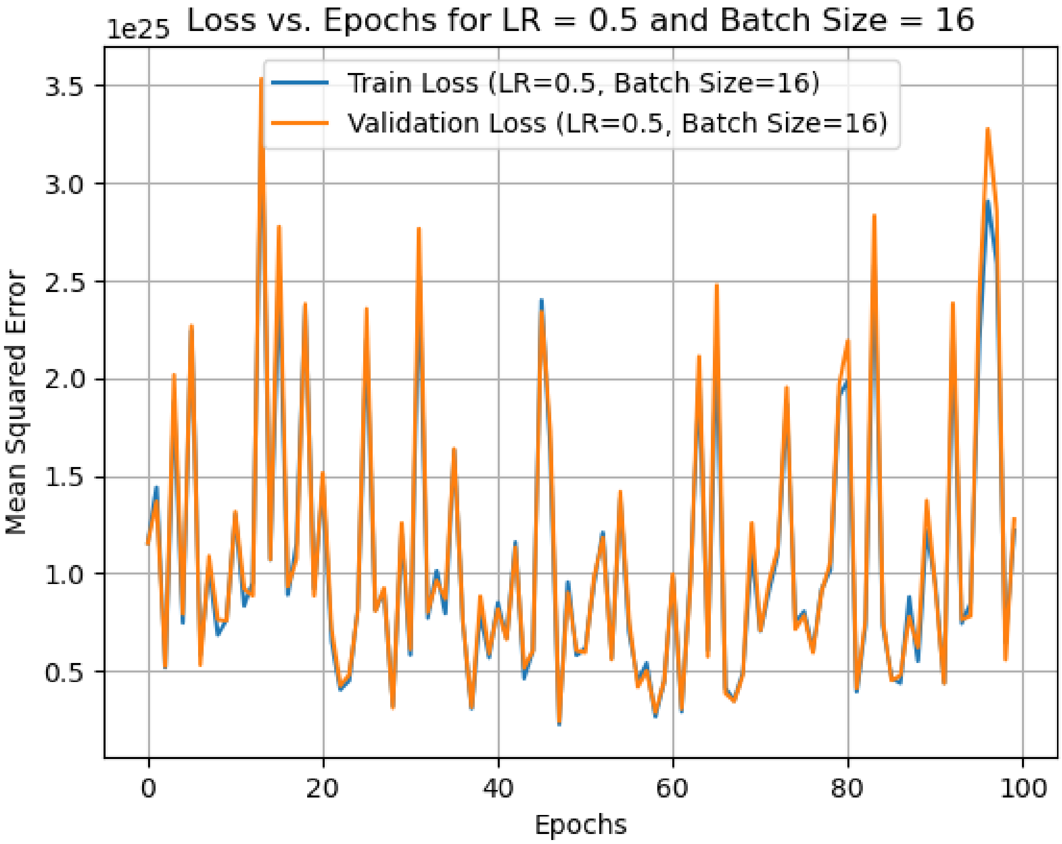
```
Training with Learning Rate = 0.5 and Batch Size = 16
Epoch 1: Train MSE = 115626827104887841027522566.0000, Validation MSE =
11552433677432699347795968.0000
Epoch 2: Train MSE = 14389589060484101488771072.0000, Validation MSE =
13680476961313086723063808.0000
Epoch 3: Train MSE = 5193934760320919423418368.0000, Validation MSE =
5255860217143920485728256.0000
Epoch 4: Train MSE = 18794830953526623465373696.0000, Validation MSE =
20168576672325881330401280.0000
Epoch 5: Train MSE = 7465488384457740300320768.0000, Validation MSE =
7929402342484271423291392.0000
Epoch 6: Train MSE = 22414392225653002527047680.0000, Validation MSE =
22671909894781485830897664.0000
Epoch 7: Train MSE = 5825926320197821723049984.0000, Validation MSE =
5317701384720244143030272.0000
Epoch 8: Train MSE = 10323371286156358929350656.0000, Validation MSE =
10889728611368202483531776.0000
Epoch 9: Train MSE = 6849864920316425516613632.0000, Validation MSE =
7621506844005329866326016.0000
```

```
Epoch 10: Train MSE = 7618719946394085317672960.0000, Validation MSE =
7582724328765556246183936.0000
Epoch 11: Train MSE = 13009335527433023952781312.0000, Validation MSE =
13163342104963262428741632.0000
Epoch 12: Train MSE = 8343312524118309830918144.0000, Validation MSE =
9211235524342734800289792.0000
Epoch 13: Train MSE = 9517638036402471297875968.0000, Validation MSE =
8850971892252232638267392.0000
Epoch 14: Train MSE = 33562485314675367632437248.0000, Validation MSE =
35313076563153681330995200.0000
Epoch 15: Train MSE = 10696680469697272668487680.0000, Validation MSE =
10704079365050751063687168.0000
Epoch 16: Train MSE = 25129842246794316397674496.0000, Validation MSE =
27747471416531200155582464.0000
Epoch 17: Train MSE = 8898295838955386563461120.0000, Validation MSE =
9322771858121678273904640.0000
Epoch 18: Train MSE = 11562618727436178937610240.0000, Validation MSE =
10732691059340830750801920.0000
Epoch 19: Train MSE = 23690311792264219361017856.0000, Validation MSE =
23771884611987201488584704.0000
Epoch 20: Train MSE = 9228900201684031340806144.0000, Validation MSE =
8849702585547511048437760.0000
Epoch 21: Train MSE = 14541198258319990942859264.0000, Validation MSE =
15144177890567870924980224.0000
Epoch 22: Train MSE = 6508339901059315049431040.0000, Validation MSE =
7158162168133443068624896.0000
Epoch 23: Train MSE = 4057430548712928726482944.0000, Validation MSE =
4256584822636697939345408.0000
Epoch 24: Train MSE = 4499357505618957133938688.0000, Validation MSE =
4795975234874136940511232.0000
Epoch 25: Train MSE = 8228321157069974670409728.0000, Validation MSE =
8120522264191882468786176.0000
Epoch 26: Train MSE = 21942411023780387217211392.0000, Validation MSE =
23536380487995648513146880.0000
Epoch 27: Train MSE = 8119764189817200317562880.0000, Validation MSE =
8075547775146238259757056.0000
Epoch 28: Train MSE = 9075935700772181183037440.0000, Validation MSE =
9241052122086820540317696.0000
Epoch 29: Train MSE = 3210088206893960325496832.0000, Validation MSE =
3107851464736893292249088.0000
Epoch 30: Train MSE = 12045132951320090573275136.0000, Validation MSE =
12578192065805370479607808.0000
Epoch 31: Train MSE = 5831790382297885020520448.0000, Validation MSE =
6123055195806087136673792.0000
Epoch 32: Train MSE = 25184750742846668072288256.0000, Validation MSE =
27646305635270158119337984.0000
Epoch 33: Train MSE = 7719115917655395511828480.0000, Validation MSE =
8007638660761003355013120.0000
```

```
Epoch 34: Train MSE = 10139917729841313270988800.0000, Validation MSE =
9638207913169653396930560.0000
```

```
Epoch 58: Train MSE = 5419855997251452938485760.0000, Validation MSE =
5032258841209157460164608.0000
Epoch 59: Train MSE = 2667601868077550069088256.0000, Validation MSE =
2895656315037763971842048.0000
Epoch 60: Train MSE = 4455171325848409281134592.0000, Validation MSE =
4471609590289254458589184.0000
Epoch 61: Train MSE = 9766186900150494777311232.0000, Validation MSE =
9946779433241031865270272.0000
Epoch 62: Train MSE = 2930501098629026717630464.0000, Validation MSE =
3072895856449393461821440.0000
Epoch 63: Train MSE = 9801047088230834708152320.0000, Validation MSE =
9436789013818466839298048.0000
Epoch 64: Train MSE = 19810830214853509634850816.0000, Validation MSE =
21096282482596419242819584.0000
Epoch 65: Train MSE = 6077956620893029358632960.0000, Validation MSE =
5735778903928824251023360.0000
Epoch 66: Train MSE = 21632322846416221459447808.0000, Validation MSE =
24741998340957941375434752.0000
Epoch 67: Train MSE = 4121456277802197574483968.0000, Validation MSE =
3848519357589876285374464.0000
Epoch 68: Train MSE = 3507919278357236028538880.0000, Validation MSE =
3463321336578542575550464.0000
Epoch 69: Train MSE = 4938897275989299531087872.0000, Validation MSE =
4801484056328475422752768.0000
Epoch 70: Train MSE = 11070288024413852078702592.0000, Validation MSE =
12584530321248375493623808.0000
Epoch 71: Train MSE = 7049734540622362117668864.0000, Validation MSE =
7080559331984514800943104.0000
Epoch 72: Train MSE = 9120701225896506990526464.0000, Validation MSE =
9572661403998986036051968.0000
Epoch 73: Train MSE = 11053333530917304583323648.0000, Validation MSE =
11304132783623132852256768.0000
Epoch 74: Train MSE = 18657220268369611927322624.0000, Validation MSE =
19517641651695781710659584.0000
Epoch 75: Train MSE = 7520606154279305042984960.0000, Validation MSE =
7150301766114367676350464.0000
Epoch 76: Train MSE = 8056788035801240074977280.0000, Validation MSE =
7853889495565243059798016.0000
Epoch 77: Train MSE = 6060527333438967659888640.0000, Validation MSE =
5956550553665000186052608.0000
Epoch 78: Train MSE = 9184745766503338626514944.0000, Validation MSE =
9084385478170181141266432.0000
Epoch 79: Train MSE = 10183493765518064607035392.0000, Validation MSE =
10584862452356144187834368.0000
Epoch 80: Train MSE = 19058264988876626708135936.0000, Validation MSE =
19680347804496847454601216.0000
Epoch 81: Train MSE = 19868489150392860325969920.0000, Validation MSE =
21893880484790154928586752.0000
```

```
Epoch 82: Train MSE = 3945195340710592534020096.0000, Validation MSE =
4128518034265971038879744.0000
Epoch 83: Train MSE = 7278440849112425820061696.0000, Validation MSE =
7612030591073353184837632.0000
Epoch 84: Train MSE = 25762224031223803015069696.0000, Validation MSE =
28316439106079239190872064.0000
Epoch 85: Train MSE = 7298343358057745804689408.0000, Validation MSE =
7520687430196840116518912.0000
Epoch 86: Train MSE = 4636493774860638880792576.0000, Validation MSE =
4519799301013051719811072.0000
Epoch 87: Train MSE = 4377155756882985392013312.0000, Validation MSE =
4784407492163544914329600.0000
Epoch 88: Train MSE = 8784552950526868029702144.0000, Validation MSE =
7789631383876836253499392.0000
Epoch 89: Train MSE = 5514247208107651123642368.0000, Validation MSE =
6211930885199641901531136.0000
Epoch 90: Train MSE = 12181658634895422201528320.0000, Validation MSE =
13716194738561390032715776.0000
Epoch 91: Train MSE = 9460230446590818282110976.0000, Validation MSE =
9173096330358688682344448.0000
Epoch 92: Train MSE = 4395138091206213940281344.0000, Validation MSE =
4374786863844172400427008.0000
Epoch 93: Train MSE = 21937368178870017593442304.0000, Validation MSE =
23804593703887293514776576.0000
Epoch 94: Train MSE = 7432793864370683734130688.0000, Validation MSE =
7623695879314824786608128.0000
Epoch 95: Train MSE = 8466851932694673994285056.0000, Validation MSE =
7832088281113179514208256.0000
Epoch 96: Train MSE = 21930006356648703159173120.0000, Validation MSE =
24193276520213068602408960.0000
Epoch 97: Train MSE = 29045637139752983554162688.0000, Validation MSE =
32747504401973568612597760.0000
Epoch 98: Train MSE = 25868895311350439480393728.0000, Validation MSE =
28529227425455160772526080.0000
Epoch 99: Train MSE = 6476973085215207314161664.0000, Validation MSE =
5582983299911756166463488.0000
Epoch 100: Train MSE = 12202953720954115042836480.0000, Validation MSE =
12775953974104524188549120.0000
```



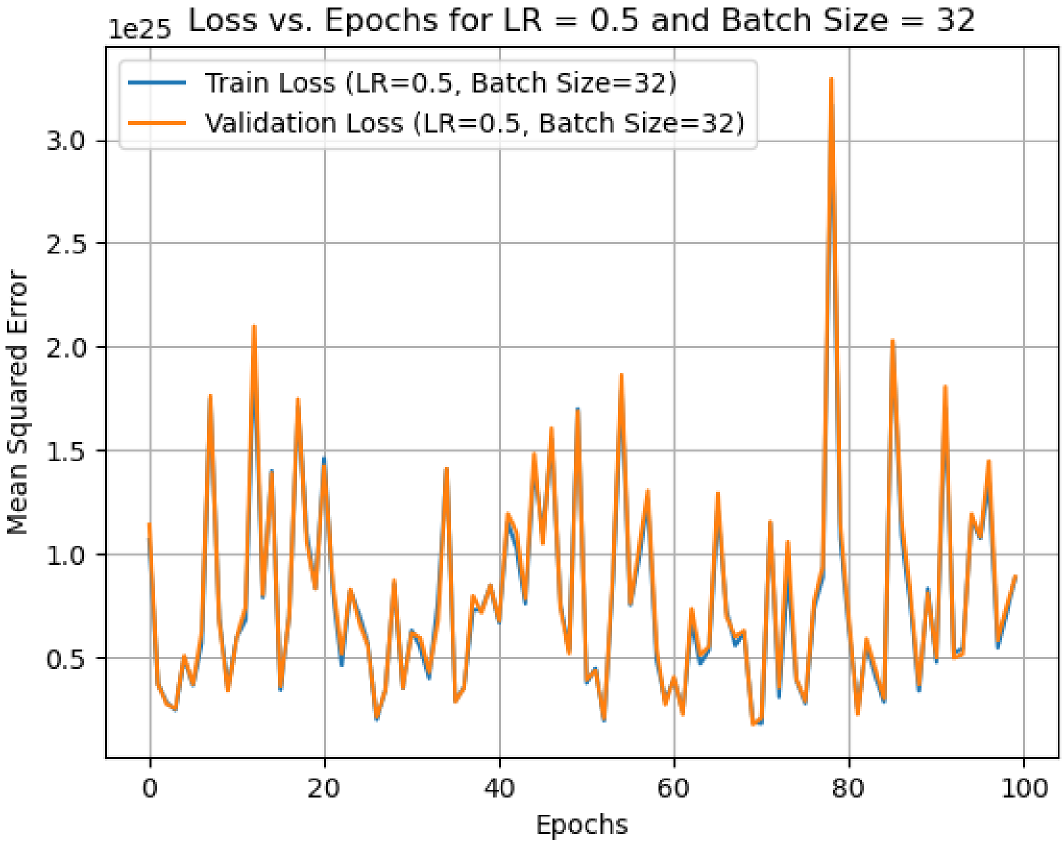
```
Training with Learning Rate = 0.5 and Batch Size = 32
Epoch 1: Train MSE = 10687801509408717276184576.0000, Validation MSE =
11412844753299381801713664.0000
Epoch 2: Train MSE = 3736839173921515803508736.0000, Validation MSE =
3716190123007885793296384.0000
Epoch 3: Train MSE = 2851381310531148001050624.0000, Validation MSE =
2775871756506302693507072.0000
Epoch 4: Train MSE = 2480585181871787330240512.0000, Validation MSE =
2547696481109097492512768.0000
Epoch 5: Train MSE = 4996332559459772799123456.0000, Validation MSE =
5098656248875150341046272.0000
Epoch 6: Train MSE = 3680033663457117966696448.0000, Validation MSE =
3732374531149044889157632.0000
Epoch 7: Train MSE = 5693882819345140724793344.0000, Validation MSE =
6299784544448372793147392.0000
Epoch 8: Train MSE = 17489373755343437215825920.0000, Validation MSE =
17644487483047637407498240.0000
Epoch 9: Train MSE = 6907841337797191703461888.0000, Validation MSE =
6746189148790390553313280.0000
```

```
Epoch 10: Train MSE = 3582846415110621496344576.0000, Validation MSE =
3425038734956595976863744.0000
Epoch 11: Train MSE = 5983694145680745422651392.0000, Validation MSE =
5977138243271596921847808.0000
Epoch 12: Train MSE = 6821128715052833910554624.0000, Validation MSE =
7433643890927201870151680.0000
Epoch 13: Train MSE = 19494491313363598790623232.0000, Validation MSE =
20994747027874623667568640.0000
Epoch 14: Train MSE = 7912091888567975835336704.0000, Validation MSE =
8040516538083289084198912.0000
Epoch 15: Train MSE = 14018102580741597091069952.0000, Validation MSE =
13908328377747255909679104.0000
Epoch 16: Train MSE = 3469217208697069380829184.0000, Validation MSE =
3644616965411818805657600.0000
Epoch 17: Train MSE = 7014751275976068306042880.0000, Validation MSE =
6996536615304647420149760.0000
Epoch 18: Train MSE = 17086889490949891459907584.0000, Validation MSE =
17480552078655726597177344.0000
Epoch 19: Train MSE = 11064758359714185290448896.0000, Validation MSE =
10598683481685601258831872.0000
Epoch 20: Train MSE = 8353758566944664306319360.0000, Validation MSE =
8341715109699494480969728.0000
Epoch 21: Train MSE = 14613011796435159952130048.0000, Validation MSE =
14248321881279048647180288.0000
Epoch 22: Train MSE = 8185479008622894745911296.0000, Validation MSE =
8588990218935448083890176.0000
Epoch 23: Train MSE = 4647880284358924002394112.0000, Validation MSE =
5202599533069248638222336.0000
Epoch 24: Train MSE = 8201550535859104871612416.0000, Validation MSE =
8293712413222794055122944.0000
Epoch 25: Train MSE = 7110443817776993552826368.0000, Validation MSE =
6729873642202342193364992.0000
Epoch 26: Train MSE = 5662890349284180084391936.0000, Validation MSE =
5562972912014163944931328.0000
Epoch 27: Train MSE = 2019557381129204563705856.0000, Validation MSE =
2150500998861905700323328.0000
Epoch 28: Train MSE = 3434036019445147807776768.0000, Validation MSE =
3427538227368898565505024.0000
Epoch 29: Train MSE = 8564091692138891027939328.0000, Validation MSE =
8750210832810550155542528.0000
Epoch 30: Train MSE = 3561365614793962867916800.0000, Validation MSE =
3566008605354728915730432.0000
Epoch 31: Train MSE = 6325942356694501628575744.0000, Validation MSE =
6193690963478864713482240.0000
Epoch 32: Train MSE = 5484620089034710940909568.0000, Validation MSE =
5955691757780721073127424.0000
Epoch 33: Train MSE = 4013817066652948787888128.0000, Validation MSE =
4301824611189644452691968.0000
```

```
Epoch 34: Train MSE = 7681708598862006531915776.0000, Validation MSE =
6792533665971039226560512.0000
Epoch 35: Train MSE = 14089112939993160394735616.0000, Validation MSE =
14122532708042724337516544.0000
Epoch 36: Train MSE = 2905664902743390582472704.0000, Validation MSE =
2886374414010052289495040.0000
Epoch 37: Train MSE = 3609597369482959476752384.0000, Validation MSE =
3559195304522557140500480.0000
Epoch 38: Train MSE = 7328247810228448210714624.0000, Validation MSE =
7984292932102948508401664.0000
Epoch 39: Train MSE = 7307300792047286244868096.0000, Validation MSE =
7184008033435799011721216.0000
Epoch 40: Train MSE = 8481838952206640437264384.0000, Validation MSE =
8497765374535653616779264.0000
Epoch 41: Train MSE = 6704593431495229478797312.0000, Validation MSE =
6801262285919427508568064.0000
Epoch 42: Train MSE = 11535297851859536265084928.0000, Validation MSE =
11942196142066257748819968.0000
Epoch 43: Train MSE = 10237997205810585964380160.0000, Validation MSE =
11053728062547894237921280.0000
Epoch 44: Train MSE = 7622570855588713996484608.0000, Validation MSE =
7862017602800901372772352.0000
Epoch 45: Train MSE = 14184629954808636487237632.0000, Validation MSE =
14830992451017776769794048.0000
Epoch 46: Train MSE = 11001302182361471314296832.0000, Validation MSE =
10529410119527782743539712.0000
Epoch 47: Train MSE = 15527858470919856300490752.0000, Validation MSE =
16067766395537687234740224.0000
Epoch 48: Train MSE = 7589615142070258610733056.0000, Validation MSE =
7424953893626561444184064.0000
Epoch 49: Train MSE = 5405247671595704483053568.0000, Validation MSE =
5226770004370221950828544.0000
Epoch 50: Train MSE = 17000417642002503618789376.0000, Validation MSE =
16891011267233457457070080.0000
Epoch 51: Train MSE = 3779979954565737873408000.0000, Validation MSE =
3935928543619897849020416.0000
Epoch 52: Train MSE = 4496971094289590800875520.0000, Validation MSE =
4386538465901589771059200.0000
Epoch 53: Train MSE = 1972409785220937480667136.0000, Validation MSE =
2079820005945717815246848.0000
Epoch 54: Train MSE = 8657497110261848468357120.0000, Validation MSE =
8993917606433339757035520.0000
Epoch 55: Train MSE = 17904423253147627615682560.0000, Validation MSE =
18650293840531292470378496.0000
Epoch 56: Train MSE = 7562718114280737364508672.0000, Validation MSE =
7649588604575788539314176.0000
Epoch 57: Train MSE = 9796463891308485502042112.0000, Validation MSE =
10300236434540644741939200.0000
```

```
Epoch 58: Train MSE = 12617049843673615164243968.0000, Validation MSE =
13057699135918020514283520.0000
Epoch 59: Train MSE = 4883438785263619461349376.0000, Validation MSE =
5428456151272861626007552.0000
Epoch 60: Train MSE = 2946840800731065279840256.0000, Validation MSE =
2756812178021032090664960.0000
Epoch 61: Train MSE = 3994766997164839728578560.0000, Validation MSE =
4070230909932730591477760.0000
Epoch 62: Train MSE = 2406562378679197095690240.0000, Validation MSE =
2279506903043636495122432.0000
Epoch 63: Train MSE = 6728982763355501653655552.0000, Validation MSE =
7367505861337940068663296.0000
Epoch 64: Train MSE = 4717647012278104750030848.0000, Validation MSE =
5142923121470415803777024.0000
Epoch 65: Train MSE = 5374372042124647859224576.0000, Validation MSE =
5551050600639116062228480.0000
Epoch 66: Train MSE = 12148190352592400315056128.0000, Validation MSE =
12931868554077499186741248.0000
Epoch 67: Train MSE = 7235670427247417183174656.0000, Validation MSE =
6977624656466984696807424.0000
Epoch 68: Train MSE = 5596639025283850706092032.0000, Validation MSE =
6008630072274479499706368.0000
Epoch 69: Train MSE = 6232237826507725682507776.0000, Validation MSE =
6318629633446539020992512.0000
Epoch 70: Train MSE = 1905311436424870493159424.0000, Validation MSE =
1793839291440437269626880.0000
Epoch 71: Train MSE = 1840861749673993613344768.0000, Validation MSE =
2099265482656692477886464.0000
Epoch 72: Train MSE = 11482121965399756987432960.0000, Validation MSE =
11568341562798601622519808.0000
Epoch 73: Train MSE = 3107095970088846127792128.0000, Validation MSE =
3580586334142767743107072.0000
Epoch 74: Train MSE = 8997880481479783048282112.0000, Validation MSE =
10587479239920921312493568.0000
Epoch 75: Train MSE = 3918173846849480722219008.0000, Validation MSE =
3966043485427520027754496.0000
Epoch 76: Train MSE = 2802935766268891872886784.0000, Validation MSE =
2906829330605230232961024.0000
Epoch 77: Train MSE = 7387253357877118598381568.0000, Validation MSE =
7726480055615710356832256.0000
Epoch 78: Train MSE = 8887629241358003577290752.0000, Validation MSE =
9393801163232828719104000.0000
Epoch 79: Train MSE = 31681017486549332137082880.0000, Validation MSE =
32927437972019157084930048.0000
Epoch 80: Train MSE = 10760824612315796439302144.0000, Validation MSE =
11494666259173425633296384.0000
Epoch 81: Train MSE = 6358257829474561742077952.0000, Validation MSE =
6703014087611209173958656.0000
```

```
Epoch 82: Train MSE = 2558061993213139169050624.0000, Validation MSE =
2295453298434394076217344.0000
Epoch 83: Train MSE = 5660746387628145493671936.0000, Validation MSE =
5916578641816289701527552.0000
Epoch 84: Train MSE = 4107361621441016596791296.0000, Validation MSE =
4447435814969023081218048.0000
Epoch 85: Train MSE = 2880384221293512203698176.0000, Validation MSE =
3050280801511135388893184.0000
Epoch 86: Train MSE = 20160512290477311914934272.0000, Validation MSE =
20284986295628941498515456.0000
Epoch 87: Train MSE = 10905831032934723328409600.0000, Validation MSE =
11635079601431117547175936.0000
Epoch 88: Train MSE = 7682235554392580485545984.0000, Validation MSE =
8167547891804787013844992.0000
Epoch 89: Train MSE = 3416293994302609853251584.0000, Validation MSE =
3716523625925776234250240.0000
Epoch 90: Train MSE = 8327882843882923027857408.0000, Validation MSE =
8129379401140231175405568.0000
Epoch 91: Train MSE = 4800817254334383955902464.0000, Validation MSE =
5012780685911294344167424.0000
Epoch 92: Train MSE = 17048971441412917329657856.0000, Validation MSE =
18085162279533088140689408.0000
Epoch 93: Train MSE = 5259182265092979810107392.0000, Validation MSE =
4996088488798638175158272.0000
Epoch 94: Train MSE = 5435518822838812605939712.0000, Validation MSE =
5188906597493073833361408.0000
Epoch 95: Train MSE = 11733512740286060107399168.0000, Validation MSE =
11932474942829956610129920.0000
Epoch 96: Train MSE = 10762253520546822027739136.0000, Validation MSE =
10831185602755727858860032.0000
Epoch 97: Train MSE = 13445694306109902181892096.0000, Validation MSE =
14491302555499567020769280.0000
Epoch 98: Train MSE = 5487031332270433187659776.0000, Validation MSE =
5847997479770489480019968.0000
Epoch 99: Train MSE = 7137854643348798307303424.0000, Validation MSE =
7461683420794392946933760.0000
Epoch 100: Train MSE = 8853066043653275661631488.0000, Validation MSE =
8903322684898876046966784.0000
```



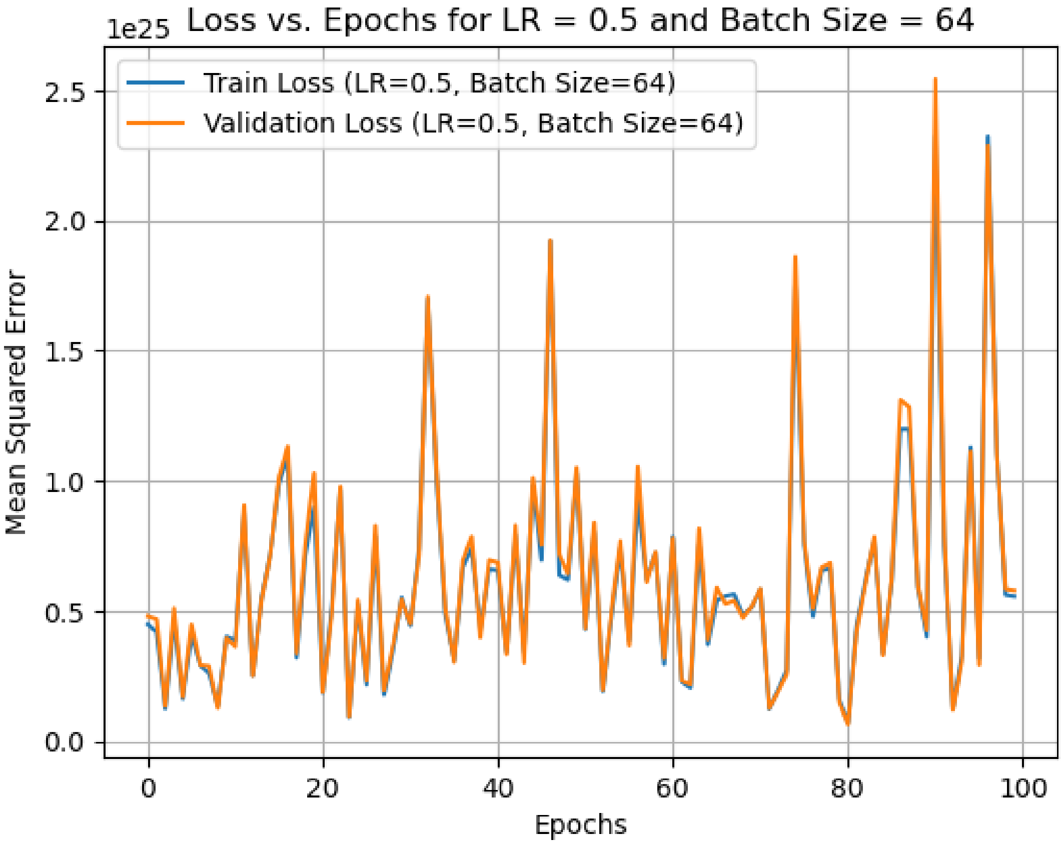
```
Training with Learning Rate = 0.5 and Batch Size = 64
Epoch 1: Train MSE = 4486460413781477290934272.0000, Validation MSE =
4807055222494834305531904.0000
Epoch 2: Train MSE = 4205790433938237544202240.0000, Validation MSE =
4696809749739472659939328.0000
Epoch 3: Train MSE = 1254079292967156338130944.0000, Validation MSE =
1355082863427729886281728.0000
Epoch 4: Train MSE = 4711490089119790382710784.0000, Validation MSE =
5118914234581316253253632.0000
Epoch 5: Train MSE = 1637593407034396592046080.0000, Validation MSE =
1706064687581527030104064.0000
Epoch 6: Train MSE = 4068431736671819547017216.0000, Validation MSE =
4477836929299605459304448.0000
Epoch 7: Train MSE = 2923495417832858607157248.0000, Validation MSE =
2928551599008350042324992.0000
Epoch 8: Train MSE = 2602287037451634600837120.0000, Validation MSE =
2889372265761789584605184.0000
Epoch 9: Train MSE = 1372756086319636306460672.0000, Validation MSE =
1278010979342781075423232.0000
```

```
Epoch 10: Train MSE = 4019344193362624325353472.0000, Validation MSE =
3958461056756161871609856.0000
Epoch 11: Train MSE = 3872748725287047388987392.0000, Validation MSE =
3625842961359612181217280.0000
Epoch 12: Train MSE = 8778251555494696688025600.0000, Validation MSE =
9068956219364643348414464.0000
Epoch 13: Train MSE = 2529965045828347836235776.0000, Validation MSE =
2521999264041718735962112.0000
Epoch 14: Train MSE = 5460550918862623022776320.0000, Validation MSE =
5400937781964118729687040.0000
Epoch 15: Train MSE = 7199725050603464608972800.0000, Validation MSE =
7165026582924667811528704.0000
Epoch 16: Train MSE = 9945051633457408692453376.0000, Validation MSE =
10161780868068249138364416.0000
Epoch 17: Train MSE = 10901554469255900842426368.0000, Validation MSE =
11325696398749938592251904.0000
Epoch 18: Train MSE = 3207776997288382247731200.0000, Validation MSE =
3388402705339395577217024.0000
Epoch 19: Train MSE = 7099088057581991800864768.0000, Validation MSE =
7565390293032331043143680.0000
Epoch 20: Train MSE = 8993960425821293995098112.0000, Validation MSE =
10306945716911999117426688.0000
Epoch 21: Train MSE = 1867457718062617148260352.0000, Validation MSE =
1885632569364993363935232.0000
Epoch 22: Train MSE = 5046707212122895601893376.0000, Validation MSE =
4798392782000360202960896.0000
Epoch 23: Train MSE = 9655424590453752499011584.0000, Validation MSE =
9777649601502429315072000.0000
Epoch 24: Train MSE = 908828407627298117779456.0000, Validation MSE =
942241740310205696049152.0000
Epoch 25: Train MSE = 5279202239109148279046144.0000, Validation MSE =
5424827129707440604446720.0000
Epoch 26: Train MSE = 2173865821106841109659648.0000, Validation MSE =
2324375814594769903943680.0000
Epoch 27: Train MSE = 7855340004099924238008320.0000, Validation MSE =
8271470196462645606875136.0000
Epoch 28: Train MSE = 1788458027311014967508992.0000, Validation MSE =
1967017765946818255912960.0000
Epoch 29: Train MSE = 3454303208681308726231040.0000, Validation MSE =
3661008748046396833136640.0000
Epoch 30: Train MSE = 5512969356726152762228736.0000, Validation MSE =
5442829511001757777395712.0000
Epoch 31: Train MSE = 4442143240634278099288064.0000, Validation MSE =
4514740675017445793071104.0000
Epoch 32: Train MSE = 7380907615709786217119744.0000, Validation MSE =
7148425760471471750119424.0000
Epoch 33: Train MSE = 16982109790917066586324992.0000, Validation MSE =
17081519186172850059870208.0000
```

```
Epoch 34: Train MSE = 10040277293302789668077568.0000, Validation MSE =
10290371653753516050087936.0000
```

```
Epoch 58: Train MSE = 6206186804924538807123968.0000, Validation MSE =
6099935284123311592701952.0000
Epoch 59: Train MSE = 7238479429243956482801664.0000, Validation MSE =
7266174990963171265609728.0000
Epoch 60: Train MSE = 2966817499863780891295744.0000, Validation MSE =
3191675906792574481858560.0000
Epoch 61: Train MSE = 7840345718134783078301696.0000, Validation MSE =
7789542226149399781703680.0000
Epoch 62: Train MSE = 2284442152129681369858048.0000, Validation MSE =
2333074816420916395769856.0000
Epoch 63: Train MSE = 2055661999083774978031616.0000, Validation MSE =
2189316851193533417127936.0000
Epoch 64: Train MSE = 7579811559843006481694720.0000, Validation MSE =
8175241685840135538606080.0000
Epoch 65: Train MSE = 3713790058935002800848896.0000, Validation MSE =
3909985991709061833293824.0000
Epoch 66: Train MSE = 5406664899905516424134656.0000, Validation MSE =
5900286830875740395798528.0000
Epoch 67: Train MSE = 5576948932192855608262656.0000, Validation MSE =
5281862255234024059437056.0000
Epoch 68: Train MSE = 5648926559018316103417856.0000, Validation MSE =
5400089393169355300143104.0000
Epoch 69: Train MSE = 4818200602362877489184768.0000, Validation MSE =
4736530563463719069679616.0000
Epoch 70: Train MSE = 5153327571752225771356160.0000, Validation MSE =
5178721474291788394004480.0000
Epoch 71: Train MSE = 5831023817717050329006080.0000, Validation MSE =
5853540247847609605029888.0000
Epoch 72: Train MSE = 1238094678195022818443264.0000, Validation MSE =
1272172748005458531319808.0000
Epoch 73: Train MSE = 1973089805162248612610048.0000, Validation MSE =
1892163021470877138026496.0000
Epoch 74: Train MSE = 2720889160888094373707776.0000, Validation MSE =
2578433403143728197009408.0000
Epoch 75: Train MSE = 17487759770722997180039168.0000, Validation MSE =
18605732380596875183521792.0000
Epoch 76: Train MSE = 7526624253573436389982208.0000, Validation MSE =
7690323801149795733602304.0000
Epoch 77: Train MSE = 4805634766461448455454720.0000, Validation MSE =
5096447765992719182200832.0000
Epoch 78: Train MSE = 6579096496867299658039296.0000, Validation MSE =
6684513750459607563632640.0000
Epoch 79: Train MSE = 6631267202285866105962496.0000, Validation MSE =
6845748457087348450000896.0000
Epoch 80: Train MSE = 1619065976573505858174976.0000, Validation MSE =
1509663603634565166399488.0000
Epoch 81: Train MSE = 706544012387464651997184.0000, Validation MSE =
650053657714162541789184.0000
```

```
Epoch 82: Train MSE = 4424434877738726849511424.0000, Validation MSE =
4254466668043115285184512.0000
Epoch 83: Train MSE = 6179472922151749731483648.0000, Validation MSE =
6039789078832939985272832.0000
Epoch 84: Train MSE = 7687286442583766203367424.0000, Validation MSE =
7842826025649468558278656.0000
Epoch 85: Train MSE = 3381377163999802287980544.0000, Validation MSE =
3301107514005288942305280.0000
Epoch 86: Train MSE = 6118124110645680595271680.0000, Validation MSE =
6224219938288791388684288.0000
Epoch 87: Train MSE = 11985909085778256318169088.0000, Validation MSE =
13112631105198003362725888.0000
Epoch 88: Train MSE = 11986770704666108694102016.0000, Validation MSE =
12833018973873203390185472.0000
Epoch 89: Train MSE = 5899008825309677079232512.0000, Validation MSE =
5951089687009233981145088.0000
Epoch 90: Train MSE = 4017646919006290121326592.0000, Validation MSE =
4266478795467303860305920.0000
Epoch 91: Train MSE = 23253378482028454652936192.0000, Validation MSE =
25444790199024466358435840.0000
Epoch 92: Train MSE = 7507695168910344614576128.0000, Validation MSE =
7876590964232730479427584.0000
Epoch 93: Train MSE = 1271478092963230966087680.0000, Validation MSE =
1208911865346620192194560.0000
Epoch 94: Train MSE = 3263155562211908071194624.0000, Validation MSE =
3053789399900155575009280.0000
Epoch 95: Train MSE = 11292131381260533838643200.0000, Validation MSE =
11118675863216054595485696.0000
Epoch 96: Train MSE = 3181693723736065984954368.0000, Validation MSE =
2939927356496636228403200.0000
Epoch 97: Train MSE = 23227349922480982564274176.0000, Validation MSE =
22864658182387334811484160.0000
Epoch 98: Train MSE = 11085793848868130256846848.0000, Validation MSE =
11017621956124798487625728.0000
Epoch 99: Train MSE = 5617507786958337644429312.0000, Validation MSE =
5839406494257575849099264.0000
Epoch 100: Train MSE = 5565879813122479467003904.0000, Validation MSE =
5787435084435524532830208.0000
```



[36]:

```
from statistics import mean


for lr, bs, tl, vl in results:
    print(f"\nSDG Training with Learning Rate = {lr} and Batch Size = {bs}")
    print('Training Loss ',mean(tl))
    print('Validation loss', mean(vl))
```

```
SDG Training with Learning Rate = 0.001 and Batch Size = 16
Training Loss  0.15852127615179173
Validation loss 0.1654144151199393

SDG Training with Learning Rate = 0.001 and Batch Size = 32
Training Loss  0.15853027966967678
Validation loss 0.165411717490708

SDG Training with Learning Rate = 0.001 and Batch Size = 64
Training Loss  0.15848726741760968
Validation loss 0.16535501629395585
```

```
SDG Training with Learning Rate = 0.01 and Batch Size = 16
Training Loss  0.145028339752951
Validation loss 0.15096249853430643

SDG Training with Learning Rate = 0.01 and Batch Size = 32
Training Loss  0.1448956140616449
Validation loss 0.15024878572365544

SDG Training with Learning Rate = 0.01 and Batch Size = 64
Training Loss  0.14454138525774712
Validation loss 0.14924321509134827

SDG Training with Learning Rate = 0.1 and Batch Size = 16
Training Loss  0.22432540088991945
Validation loss 0.23866149983618354

SDG Training with Learning Rate = 0.1 and Batch Size = 32
Training Loss  0.23068578032093606
Validation loss 0.2404156815959469

SDG Training with Learning Rate = 0.1 and Batch Size = 64
Training Loss  0.25037739764479183
Validation loss 0.2544506819944055

SDG Training with Learning Rate = 0.5 and Batch Size = 16
Training Loss  1.0667486470150418e+25
Validation loss 1.1049255847854963e+25

SDG Training with Learning Rate = 0.5 and Batch Size = 32
Training Loss  7.872089265694358e+24
Validation loss 8.08473575043729e+24

SDG Training with Learning Rate = 0.5 and Batch Size = 64
Training Loss  6.150807480104084e+24
Validation loss 6.324748846294972e+24
```

12 Overall Summary

1. **Best Performance:** The combination of a learning rate of **0.001** and a batch size of **64** resulted in the most stable accuracy and decreasing loss.
2. **Batch Size Impact:** Larger batch sizes (like 32 and 64) generally provided better stability and lower loss, especially with lower learning rates.
3. **Learning Rates:** Higher learning rates (0.1 and 0.5) led to inconsistent performance and potential overfitting. A lower learning rate (0.01) resulted in low accuracy for all batch sizes.

12.1 Recommendations

- Use learning rates between **0.001** and **0.1** with larger batch sizes (32-64) for improved performance.
- Consider regularization to prevent overfitting with higher learning rates and smaller batches.

In summary, adjusting learning rates and batch sizes can enhance model effectiveness and stability.

13 F

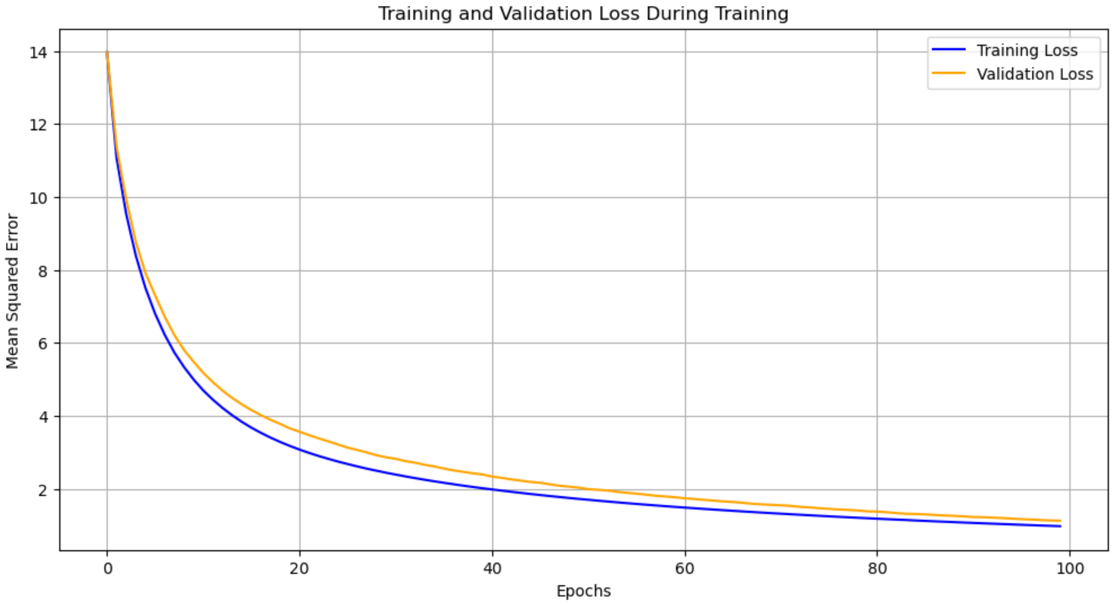
[29]:

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
import numpy as np
import matplotlib.pyplot as plt

# Create polynomial features
degree = 2  # Adjust the degree as necessary
poly = PolynomialFeatures(degree=degree)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)

# Print shapes of the transformed features
print(f'Shape of X_train_poly: {X_train_poly.shape}')
print(f'Shape of y_train: {y_train.shape}')
print(f'Shape of X_test_poly: {X_test_poly.shape}')
print(f'Shape of y_test: {y_test.shape}')

# Function to perform polynomial regression using SGD
def polynomial_regression_sgd(X, y, learning_rate=0.001, n_epochs=100,␣
 ↪batch_size=64):
    m, n = X.shape  # Number of samples and number of features
    weights = np.zeros(n)  # Initialize weights with the same number as features
    train_losses = []
    val_losses = []

    for epoch in range(n_epochs):
        indices = np.random.permutation(m)
        X_shuffled = X[indices]
        y_shuffled = y.iloc[indices].values.flatten()  # Ensure y_shuffled is a␣
 ↪1D array

        for i in range(0, m, batch_size):
            X_batch = X_shuffled[i:i + batch_size]
            y_batch = y_shuffled[i:i + batch_size]

            # Ensure the batch size doesn't exceed
            if X_batch.shape[0] < batch_size:
                continue
            gradients = -2 / X_batch.shape[0] * X_batch.T.dot(y_batch - X_batch.
 ↪dot(weights))
            weights -= learning_rate * gradients

        # Compute training loss
        train_loss = mean_squared_error(y.values.flatten(), X.dot(weights))  #␣
 ↪Ensure y is 1D
        train_losses.append(train_loss)

        # Compute validation loss
        val_loss = mean_squared_error(y_test.values.flatten(), X_test_poly.
 ↪dot(weights))  # Ensure y_test is 1D
        val_losses.append(val_loss)

        # Print iteration and losses
        print(f'Epoch: {epoch + 1}/{n_epochs}, Training Loss: {train_loss:.4f},␣
 ↪Validation Loss: {val_loss:.4f}')

    # Plot training and validation loss
    plt.figure(figsize=(12, 6))
    plt.plot(train_losses, label='Training Loss', color='blue')
    plt.plot(val_losses, label='Validation Loss', color='orange')
    plt.title('Training and Validation Loss During Training')
    plt.xlabel('Epochs')
    plt.ylabel('Mean Squared Error')
    plt.legend()
    plt.grid()
    plt.show()

    return weights, train_losses, val_losses

# Train the model with specified learning rate and batch size
weights_poly_sgd, train_losses, val_losses =␣
 ↪polynomial_regression_sgd(X_train_poly, y_train)
```

```
Shape of X_train_poly: (1461, 55)
Shape of y_train: (1461, 1)
Shape of X_test_poly: (488, 55)
Shape of y_test: (488, 1)
Epoch: 1/100, Training Loss: 13.9651, Validation Loss: 13.9654
Epoch: 2/100, Training Loss: 11.0646, Validation Loss: 11.3721
Epoch: 3/100, Training Loss: 9.5321, Validation Loss: 9.9316
Epoch: 4/100, Training Loss: 8.3901, Validation Loss: 8.7708
Epoch: 5/100, Training Loss: 7.5053, Validation Loss: 7.9168
Epoch: 6/100, Training Loss: 6.8041, Validation Loss: 7.3073
Epoch: 7/100, Training Loss: 6.2232, Validation Loss: 6.7213
```

```
Epoch: 8/100, Training Loss: 5.7402, Validation Loss: 6.2174
Epoch: 9/100, Training Loss: 5.3421, Validation Loss: 5.8116
Epoch: 10/100, Training Loss: 4.9993, Validation Loss: 5.4852
Epoch: 11/100, Training Loss: 4.7009, Validation Loss: 5.1855
Epoch: 12/100, Training Loss: 4.4461, Validation Loss: 4.9270
Epoch: 13/100, Training Loss: 4.2209, Validation Loss: 4.6973
Epoch: 14/100, Training Loss: 4.0203, Validation Loss: 4.4957
Epoch: 15/100, Training Loss: 3.8407, Validation Loss: 4.3201
Epoch: 16/100, Training Loss: 3.6798, Validation Loss: 4.1620
Epoch: 17/100, Training Loss: 3.5365, Validation Loss: 4.0221
Epoch: 18/100, Training Loss: 3.4057, Validation Loss: 3.8945
Epoch: 19/100, Training Loss: 3.2872, Validation Loss: 3.7817
Epoch: 20/100, Training Loss: 3.1787, Validation Loss: 3.6605
Epoch: 21/100, Training Loss: 3.0792, Validation Loss: 3.5678
Epoch: 22/100, Training Loss: 2.9872, Validation Loss: 3.4745
Epoch: 23/100, Training Loss: 2.9023, Validation Loss: 3.3861
Epoch: 24/100, Training Loss: 2.8247, Validation Loss: 3.3043
Epoch: 25/100, Training Loss: 2.7507, Validation Loss: 3.2166
Epoch: 26/100, Training Loss: 2.6828, Validation Loss: 3.1329
Epoch: 27/100, Training Loss: 2.6176, Validation Loss: 3.0693
Epoch: 28/100, Training Loss: 2.5561, Validation Loss: 2.9996
Epoch: 29/100, Training Loss: 2.4999, Validation Loss: 2.9250
Epoch: 30/100, Training Loss: 2.4450, Validation Loss: 2.8666
Epoch: 31/100, Training Loss: 2.3937, Validation Loss: 2.8254
Epoch: 32/100, Training Loss: 2.3441, Validation Loss: 2.7622
Epoch: 33/100, Training Loss: 2.2975, Validation Loss: 2.7205
Epoch: 34/100, Training Loss: 2.2525, Validation Loss: 2.6634
Epoch: 35/100, Training Loss: 2.2098, Validation Loss: 2.6175
Epoch: 36/100, Training Loss: 2.1689, Validation Loss: 2.5588
Epoch: 37/100, Training Loss: 2.1297, Validation Loss: 2.5093
Epoch: 38/100, Training Loss: 2.0912, Validation Loss: 2.4678
Epoch: 39/100, Training Loss: 2.0547, Validation Loss: 2.4297
Epoch: 40/100, Training Loss: 2.0202, Validation Loss: 2.3969
Epoch: 41/100, Training Loss: 1.9873, Validation Loss: 2.3393
Epoch: 42/100, Training Loss: 1.9538, Validation Loss: 2.3030
Epoch: 43/100, Training Loss: 1.9226, Validation Loss: 2.2621
Epoch: 44/100, Training Loss: 1.8923, Validation Loss: 2.2278
Epoch: 45/100, Training Loss: 1.8623, Validation Loss: 2.1929
Epoch: 46/100, Training Loss: 1.8335, Validation Loss: 2.1696
Epoch: 47/100, Training Loss: 1.8060, Validation Loss: 2.1276
Epoch: 48/100, Training Loss: 1.7798, Validation Loss: 2.0876
Epoch: 49/100, Training Loss: 1.7528, Validation Loss: 2.0639
Epoch: 50/100, Training Loss: 1.7279, Validation Loss: 2.0360
Epoch: 51/100, Training Loss: 1.7043, Validation Loss: 1.9955
Epoch: 52/100, Training Loss: 1.6793, Validation Loss: 1.9770
Epoch: 53/100, Training Loss: 1.6562, Validation Loss: 1.9563
Epoch: 54/100, Training Loss: 1.6336, Validation Loss: 1.9225
Epoch: 55/100, Training Loss: 1.6115, Validation Loss: 1.8949
```

```
Epoch: 56/100, Training Loss: 1.5899, Validation Loss: 1.8714
Epoch: 57/100, Training Loss: 1.5688, Validation Loss: 1.8435
Epoch: 58/100, Training Loss: 1.5489, Validation Loss: 1.8150
Epoch: 59/100, Training Loss: 1.5284, Validation Loss: 1.7951
Epoch: 60/100, Training Loss: 1.5088, Validation Loss: 1.7713
Epoch: 61/100, Training Loss: 1.4902, Validation Loss: 1.7452
Epoch: 62/100, Training Loss: 1.4719, Validation Loss: 1.7247
Epoch: 63/100, Training Loss: 1.4537, Validation Loss: 1.7019
Epoch: 64/100, Training Loss: 1.4359, Validation Loss: 1.6814
Epoch: 65/100, Training Loss: 1.4188, Validation Loss: 1.6586
Epoch: 66/100, Training Loss: 1.4020, Validation Loss: 1.6439
Epoch: 67/100, Training Loss: 1.3852, Validation Loss: 1.6215
Epoch: 68/100, Training Loss: 1.3692, Validation Loss: 1.5955
Epoch: 69/100, Training Loss: 1.3529, Validation Loss: 1.5785
Epoch: 70/100, Training Loss: 1.3377, Validation Loss: 1.5649
Epoch: 71/100, Training Loss: 1.3229, Validation Loss: 1.5538
Epoch: 72/100, Training Loss: 1.3082, Validation Loss: 1.5360
Epoch: 73/100, Training Loss: 1.2934, Validation Loss: 1.5101
Epoch: 74/100, Training Loss: 1.2789, Validation Loss: 1.4945
Epoch: 75/100, Training Loss: 1.2645, Validation Loss: 1.4722
Epoch: 76/100, Training Loss: 1.2504, Validation Loss: 1.4548
Epoch: 77/100, Training Loss: 1.2366, Validation Loss: 1.4384
Epoch: 78/100, Training Loss: 1.2235, Validation Loss: 1.4257
Epoch: 79/100, Training Loss: 1.2106, Validation Loss: 1.4103
Epoch: 80/100, Training Loss: 1.1983, Validation Loss: 1.3863
Epoch: 81/100, Training Loss: 1.1853, Validation Loss: 1.3808
Epoch: 82/100, Training Loss: 1.1726, Validation Loss: 1.3618
Epoch: 83/100, Training Loss: 1.1610, Validation Loss: 1.3397
Epoch: 84/100, Training Loss: 1.1501, Validation Loss: 1.3208
Epoch: 85/100, Training Loss: 1.1365, Validation Loss: 1.3144
Epoch: 86/100, Training Loss: 1.1249, Validation Loss: 1.3028
Epoch: 87/100, Training Loss: 1.1134, Validation Loss: 1.2873
Epoch: 88/100, Training Loss: 1.1021, Validation Loss: 1.2732
Epoch: 89/100, Training Loss: 1.0911, Validation Loss: 1.2632
Epoch: 90/100, Training Loss: 1.0801, Validation Loss: 1.2486
Epoch: 91/100, Training Loss: 1.0695, Validation Loss: 1.2324
Epoch: 92/100, Training Loss: 1.0592, Validation Loss: 1.2265
Epoch: 93/100, Training Loss: 1.0487, Validation Loss: 1.2143
Epoch: 94/100, Training Loss: 1.0388, Validation Loss: 1.2035
Epoch: 95/100, Training Loss: 1.0283, Validation Loss: 1.1863
Epoch: 96/100, Training Loss: 1.0184, Validation Loss: 1.1738
Epoch: 97/100, Training Loss: 1.0086, Validation Loss: 1.1624
Epoch: 98/100, Training Loss: 0.9989, Validation Loss: 1.1486
Epoch: 99/100, Training Loss: 0.9898, Validation Loss: 1.1368
Epoch: 100/100, Training Loss: 0.9805, Validation Loss: 1.1305
```



[37]:
```
print('polynomial_regression_sgd')
print('train_loss', mean(train_losses))
print('Validation loss', mean(val_losses))
```

```
polynomial_regression_sgd
train_loss 2.5000097106697967
Validation loss 2.8129829480126896
```

13.0.1 Exploring Overfitting and Underfitting Using Validation Loss

Observations:

1. **Both Training and Validation Loss Decrease**:
   - The training loss (blue curve) and validation loss (orange curve) both decrease consistently over the epochs.
   - This indicates that the model is learning from the data and improving its performance on both the training and validation sets.
2. **No Significant Gap Between Training and Validation Loss**:
   - The gap between training and validation loss is relatively small throughout the training process, suggesting that the model is generalizing well to unseen data (validation set).
   - This suggests that **overfitting** is **not a significant concern** in the current model, as the model is able to perform similarly on both the training and validation sets.

Conclusion:

- **No Overfitting**: Since the validation loss decreases in a similar pattern to the training loss, there is no sign of overfitting.
- **No Underfitting**: Both training and validation losses steadily decrease, indicating that the model is not underfitting and is successfully capturing patterns in the data.

Final Answer: Based on the training and validation loss curves, the model does not appear to be overfitting or underfitting. The steady decline in both losses without a significant gap between them suggests that the model is well-fitted to the data and generalizes well to unseen data.

13.1 Recommended Model

Given the results, I recommend choosing either **Ridge Regression** with **alpha=0.1** or **Elastic Net** with **alpha=0.001** and **l1_ratio=0.2 or 0.5** as the best models. These options exhibit a combination of low **Test Mean Squared Error (MSE)** and reasonably consistent performance across the training and validation datasets.

- **Ridge Regression (alpha=0.1)**:
  - **Pros**:
    - More interpretable model, making it easier to understand the influence of individual features.
    - Effective in preventing overfitting, especially when multicollinearity is present.
- **Elastic Net (alpha=0.001, l1_ratio=0.2 or 0.5)**:
  - **Pros**:
    - Balances feature selection and regularization, which can be beneficial if the number of features is high.
    - Can potentially outperform Ridge Regression by combining the benefits of both Lasso and Ridge regularization.

In summary, if interpretability is a primary concern, **Ridge Regression** may be preferable. However, if the model needs to handle a larger set of features while still controlling for overfitting, **Elastic Net** is a strong alternative.

14 G

[40]:
```python
# Step 1: Train the Ridge Regression Model
ridge_model = Ridge(alpha=0.1)
ridge_model.fit(X_train, y_train)

# Step 2: Make Predictions on the Test Data
y_pred = ridge_model.predict(X_test)

# Step 3: Evaluate Performance
test_mse = mean_squared_error(y_test, y_pred)
test_r2 = r2_score(y_test, y_pred)

# Print the evaluation results
print(f"Test MSE: {test_mse:.4f}")
print(f"Test R²: {test_r2:.4f}")
```

```
Test MSE: 0.1471
Test R²: 0.8902
```

14.0.1 Findings

The Ridge Regression model with alpha = 0.1 achieved:

- **Test MSE**: **0.1471**
- **Test R²**: **0.8902**

This indicates that the model has strong predictive performance, explaining about 89% of the variance in the target variable, with a relatively low average squared difference between predicted and actual values.

14.0.2 Further Exploration to Increase Performance

1. **Hyperparameter Tuning**:
   - Use grid or random search to explore a broader range of alpha values.
   - Implement k-fold cross-validation for robust performance evaluation.
2. **Feature Engineering**:
   - Create polynomial features or interaction terms.
   - Apply feature selection techniques to enhance model inputs.
3. **Data Preprocessing**:
   - Ensure all features are properly scaled.
   - Handle any missing values to improve model accuracy.
4. **Model Complexity**:
   - Explore ensemble methods (e.g., Random Forests, Gradient Boosting).
   - Consider other regularization techniques like Lasso or Elastic Net.
5. **Residual Analysis**:
   - Analyze residuals to identify patterns that indicate model weaknesses.
6. **Increasing Data**:
   - If feasible, gather more training data or augment the existing dataset for better generalization.

By implementing these strategies, the model's predictive accuracy can potentially be enhanced even further.