Forecasting and Evaluating Property Values in Melbourne

By Michael Le

Melbourne

This is my first Kaggle data-analysis and data-modelling project, employing various Machine Learning techniques including Supervised, Unsupervised Learning. Including exploration to predict house prices in Melbourne, leveraging a dataset spanning the years from 2016-2017.

The following dataset consists of

Rooms: Number of rooms Price: Price in dolla

Method: S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN - sold not disclosed; NB## id; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS - sold after auction price not disclosed. N/A - price or hest### bid not aabl## e.

Type: br - bedroom(s); h - house,cottage,villa, semi,terrace; u - unit, duplex; t - townhouse; dev site - development site

othernt## ial.

Sellel astG: ate## D Agente: D## ate sold

Dististanc## e from CBD

Regionname: General Region (West, North Weth, Nort## h east …etc)

Propertycount: Nurtiesber of propsthat ex ist ## in thesburb.

Bnumber rbem2 : Scraped #eoms (from d## ifferent sumber Bathroom: N## uar: Nu Bathrooms

Caber of carspots###

e: Land Size

Bui###ldingArea: Building Size

CouncilAreal for the areae | --- |

Note that the columns for Bedroom2 has changed to Bedroom and SellerG to Seller to avoid confusion due to typo errors.

```
#This is to add in raw button and  Python 3 compat
#Ensure the Notebook has added support for raw_input and %debug, as of
1.0. in Github
```

```
import sys
if sys.version_info[0] >= 3:
    raw_input = input
```

Step 1. Loading the Dataset for the Melbourne House Prices

```
#Importing the Pandas and Numpy Package
import pandas as pd
import numpy as np

housing_pd =
pd.read_csv("Desktop/melbourne_house_prices/Melbourne_housing_FULL.csv
")
housing_pd.head()
```

```
       Suburb             Address  Rooms Type      Price Method
Seller  \
0  Abbotsford       68 Studley St      2    h        NaN     SS
Jellis
1  Abbotsford        85 Turner St      2    h  1480000.0      S
Biggin
2  Abbotsford     25 Bloomburg St      2    h  1035000.0      S
Biggin
3  Abbotsford  18/659 Victoria St      3    u        NaN     VB
Rounds
4  Abbotsford        5 Charles St      3    h  1465000.0     SP
Biggin

        Date  Distance  Postcode  ...  Bathroom  Car  Landsize
BuildingArea  \
0  3/09/2016       2.5    3067.0  ...       1.0  1.0     126.0
NaN
1  3/12/2016       2.5    3067.0  ...       1.0  1.0     202.0
NaN
2  4/02/2016       2.5    3067.0  ...       1.0  0.0     156.0
79.0
3  4/02/2016       2.5    3067.0  ...       2.0  1.0       0.0
NaN
4  4/03/2017       2.5    3067.0  ...       2.0  0.0     134.0
150.0

   YearBuilt         CouncilArea Lattitude  Longtitude
Regionname  \
0        NaN  Yarra City Council  -37.8014    144.9958  Northern
Metropolitan
1        NaN  Yarra City Council  -37.7996    144.9984  Northern
```

```
Metropolitan
2     1900.0  Yarra City Council  -37.8079    144.9934  Northern
Metropolitan
3        NaN  Yarra City Council  -37.8114    145.0116  Northern
Metropolitan
4     1900.0  Yarra City Council  -37.8093    144.9944  Northern
Metropolitan

  Propertycount
0        4019.0
1        4019.0
2        4019.0
3        4019.0
4        4019.0

[5 rows x 21 columns]
```

```
housing_pd.shape
```

```
(34857, 21)
```

Step 2. Data Cleaning

Re-order the columns of the data-frame

```
housing_pd = housing_pd[['Price','Suburb', 'Address', 'Rooms',
'Type','Method', 'Seller',
       'Date', 'Distance', 'Postcode', 'Bedroom', 'Bathroom', 'Car',
       'Landsize', 'BuildingArea', 'YearBuilt', 'CouncilArea',
'Lattitude',
       'Longtitude', 'Regionname', 'Propertycount']]
```

Check data info

```
housing_pd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34857 entries, 0 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Price          27247 non-null  float64
 1   Suburb         34857 non-null  object
 2   Address        34857 non-null  object
 3   Rooms          34857 non-null  int64
 4   Type           34857 non-null  object
 5   Method         34857 non-null  object
 6   Seller         34857 non-null  object
```

```
 7   Date           34857 non-null  object
 8   Distance       34856 non-null  float64
 9   Postcode       34856 non-null  float64
 10  Bedroom        26640 non-null  float64
 11  Bathroom       26631 non-null  float64
 12  Car            26129 non-null  float64
 13  Landsize       23047 non-null  float64
 14  BuildingArea   13742 non-null  float64
 15  YearBuilt      15551 non-null  float64
 16  CouncilArea    34854 non-null  object
 17  Lattitude      26881 non-null  float64
 18  Longtitude     26881 non-null  float64
 19  Regionname     34854 non-null  object
 20  Propertycount  34854 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 5.6+ MB
```

Count how many null values for each of the columns in the housing dataset.

```
housing_pd.isnull().sum()
```

```
Price             7610
Suburb               0
Address              0
Rooms                0
Type                 0
Method               0
Seller               0
Date                 0
Distance             1
Postcode             1
Bedroom           8217
Bathroom          8226
Car               8728
Landsize         11810
BuildingArea     21115
YearBuilt        19306
CouncilArea          3
Lattitude         7976
Longtitude        7976
Regionname           3
Propertycount        3
dtype: int64
```

Check if there are any duplicated rows?

```
housing_pd.duplicated().any()
```

```
True
```

Remove duplicated rows

```
housing_pd = housing_pd.drop_duplicates()
housing_pd
```

```
           Price       Suburb             Address  Rooms Type Method  \
0            NaN  Abbotsford       68 Studley St      2    h     SS   
1      1480000.0  Abbotsford        85 Turner St      2    h      S   
2      1035000.0  Abbotsford     25 Bloomburg St      2    h      S   
3            NaN  Abbotsford  18/659 Victoria St      3    u     VB   
4      1465000.0  Abbotsford        5 Charles St      3    h     SP   
...          ...          ...                 ...    ...  ...    ...   
34852  1480000.0  Yarraville         13 Burns St      4    h     PI   
34853   888000.0  Yarraville       29A Murray St      2    h     SP   
34854   705000.0  Yarraville      147A Severn St      2    t      S   
34855  1140000.0  Yarraville    12/37 Stephen St      3    h     SP   
34856  1020000.0  Yarraville    3 Tarrengower St      2    h     PI   

              Seller        Date  Distance  Postcode  ...  Bathroom 
Car  \
0             Jellis   3/09/2016       2.5    3067.0  ...       1.0 
1.0   
1             Biggin   3/12/2016       2.5    3067.0  ...       1.0 
1.0   
2             Biggin   4/02/2016       2.5    3067.0  ...       1.0 
0.0   
3             Rounds   4/02/2016       2.5    3067.0  ...       2.0 
1.0   
4             Biggin   4/03/2017       2.5    3067.0  ...       2.0 
0.0   
...              ...         ...       ...       ...  ...       ...  .
..   
34852            Jas  24/02/2018       6.3    3013.0  ...       1.0 
3.0   
34853        Sweeney  24/02/2018       6.3    3013.0  ...       2.0 
1.0   
34854            Jas  24/02/2018       6.3    3013.0  ...       1.0 
2.0   
34855  hockingstuart  24/02/2018       6.3    3013.0  ...       NaN 
NaN   
34856             RW  24/02/2018       6.3    3013.0  ...       1.0 
0.0   

       Landsize  BuildingArea  YearBuilt               CouncilArea 
Lattitude  \
0         126.0           NaN        NaN        Yarra City Council -
```

```
37.80140
1          202.0            NaN         NaN        Yarra City Council -
37.79960
2          156.0           79.0      1900.0        Yarra City Council -
37.80790
3            0.0            NaN         NaN        Yarra City Council -
37.81140
4          134.0          150.0      1900.0        Yarra City Council -
37.80930
...          ...            ...         ...                       ...
...
34852      593.0            NaN         NaN  Maribyrnong City Council -
37.81053
34853       98.0          104.0      2018.0  Maribyrnong City Council -
37.81551
34854      220.0          120.0      2000.0  Maribyrnong City Council -
37.82286
34855        NaN            NaN         NaN  Maribyrnong City Council
NaN
34856      250.0          103.0      1930.0  Maribyrnong City Council -
37.81810

       Longtitude             Regionname Propertycount
0       144.99580  Northern Metropolitan        4019.0
1       144.99840  Northern Metropolitan        4019.0
2       144.99340  Northern Metropolitan        4019.0
3       145.01160  Northern Metropolitan        4019.0
4       144.99440  Northern Metropolitan        4019.0
...           ...                    ...           ...
34852   144.88467   Western Metropolitan        6543.0
34853   144.88826   Western Metropolitan        6543.0
34854   144.87856   Western Metropolitan        6543.0
34855         NaN   Western Metropolitan        6543.0
34856   144.89351   Western Metropolitan        6543.0

[34856 rows x 21 columns]
```

One approach to cleaning data: Get rid off all rows containing NaN alphameric and numeric values.

```
housing_pd = housing_pd.dropna()
housing_pd.info()

<class 'pandas.core.frame.DataFrame'>
Index: 8887 entries, 2 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Price          8887 non-null   float64
```

```
 1   Suburb         8887 non-null   object
 2   Address        8887 non-null   object
 3   Rooms          8887 non-null   int64
 4   Type           8887 non-null   object
 5   Method         8887 non-null   object
 6   Seller         8887 non-null   object
 7   Date           8887 non-null   object
 8   Distance       8887 non-null   float64
 9   Postcode       8887 non-null   float64
 10  Bedroom        8887 non-null   float64
 11  Bathroom       8887 non-null   float64
 12  Car            8887 non-null   float64
 13  Landsize       8887 non-null   float64
 14  BuildingArea   8887 non-null   float64
 15  YearBuilt      8887 non-null   float64
 16  CouncilArea    8887 non-null   object
 17  Lattitude      8887 non-null   float64
 18  Longtitude     8887 non-null   float64
 19  Regionname     8887 non-null   object
 20  Propertycount  8887 non-null   float64
dtypes: float64(12), int64(1), object(8)
memory usage: 1.5+ MB
```

May add other methods to cleaning data. Which would impact the results of the prediction on house prices in Melbourne.

```

```

Change the columns Bedroom, Bathroom, Car, YearBuilt and Property Count into the right metrics.

```
housing_pd['Bedroom'] = housing_pd['Bedroom'].astype(int)
housing_pd['Bathroom'] = housing_pd['Bathroom'].astype(int)
housing_pd['Car'] = housing_pd['Car'].astype(int)
housing_pd['YearBuilt'] = housing_pd['YearBuilt'].astype(int)
housing_pd['Propertycount'] = housing_pd['Propertycount'].astype(int)
```

```
C:\Users\Michael Le\AppData\Local\Temp\
ipykernel_40248\1123500922.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
```

```
  housing_pd['Bedroom'] = housing_pd['Bedroom'].astype(int)
C:\Users\Michael Le\AppData\Local\Temp\
ipykernel_40248\1123500922.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  housing_pd['Bathroom'] = housing_pd['Bathroom'].astype(int)
C:\Users\Michael Le\AppData\Local\Temp\
ipykernel_40248\1123500922.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  housing_pd['Car'] = housing_pd['Car'].astype(int)
C:\Users\Michael Le\AppData\Local\Temp\
ipykernel_40248\1123500922.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  housing_pd['YearBuilt'] = housing_pd['YearBuilt'].astype(int)
C:\Users\Michael Le\AppData\Local\Temp\
ipykernel_40248\1123500922.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  housing_pd['Propertycount'] =
housing_pd['Propertycount'].astype(int)
```

Check the info

```
housing_pd.info()

<class 'pandas.core.frame.DataFrame'>
Index: 8887 entries, 2 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Price          8887 non-null   float64
 1   Suburb         8887 non-null   object
```

```
 2   Address        8887 non-null   object
 3   Rooms          8887 non-null   int64
 4   Type           8887 non-null   object
 5   Method         8887 non-null   object
 6   Seller         8887 non-null   object
 7   Date           8887 non-null   object
 8   Distance       8887 non-null   float64
 9   Postcode       8887 non-null   float64
 10  Bedroom        8887 non-null   int32
 11  Bathroom       8887 non-null   int32
 12  Car            8887 non-null   int32
 13  Landsize       8887 non-null   float64
 14  BuildingArea   8887 non-null   float64
 15  YearBuilt      8887 non-null   int32
 16  CouncilArea    8887 non-null   object
 17  Lattitude      8887 non-null   float64
 18  Longtitude     8887 non-null   float64
 19  Regionname     8887 non-null   object
 20  Propertycount  8887 non-null   int32
dtypes: float64(7), int32(5), int64(1), object(8)
memory usage: 1.3+ MB
```

Refer to https://stackoverflow.com/questions/15891038/change-column-type-in-pandas

Check the head of the cleaned dataframe for the Melbourne housing data

```
housing_pd.head()

        Price      Suburb          Address  Rooms Type Method  Seller
\
2   1035000.0  Abbotsford  25 Bloomburg St      2    h      S  Biggin

4   1465000.0  Abbotsford     5 Charles St      3    h     SP  Biggin

6   1600000.0  Abbotsford      55a Park St      4    h     VB  Nelson

11  1876000.0  Abbotsford     124 Yarra St      3    h      S  Nelson

14  1636000.0  Abbotsford    98 Charles St      2    h      S  Nelson


         Date  Distance  Postcode  ...  Bathroom  Car  Landsize
BuildingArea  \
2   4/02/2016       2.5    3067.0  ...         1    0     156.0
```

```
79.0
4   4/03/2017       2.5    3067.0  ...         2    0      134.0
150.0
6   4/06/2016       2.5    3067.0  ...         1    2      120.0
142.0
11  7/05/2016       2.5    3067.0  ...         2    0      245.0
210.0
14  8/10/2016       2.5    3067.0  ...         1    2      256.0
107.0

    YearBuilt         CouncilArea Lattitude  Longtitude  \
2        1900  Yarra City Council  -37.8079    144.9934
4        1900  Yarra City Council  -37.8093    144.9944
6        2014  Yarra City Council  -37.8072    144.9941
11       1910  Yarra City Council  -37.8024    144.9993
14       1890  Yarra City Council  -37.8060    144.9954

               Regionname Propertycount
2   Northern Metropolitan          4019
4   Northern Metropolitan          4019
6   Northern Metropolitan          4019
11  Northern Metropolitan          4019
14  Northern Metropolitan          4019

[5 rows x 21 columns]
```

Getting rid of columns we do not need (might be useful for Step 3.)

```
housing_pd = housing_pd.drop(['Address','Date'],axis=1)

#Check the head to see if we got rid of Address and Data
housing_pd.head()

        Price      Suburb  Rooms Type Method  Seller  Distance
Postcode  \
2   1035000.0  Abbotsford      2    h      S  Biggin       2.5
3067.0
4   1465000.0  Abbotsford      3    h     SP  Biggin       2.5
3067.0
6   1600000.0  Abbotsford      4    h     VB  Nelson       2.5
3067.0
11  1876000.0  Abbotsford      3    h      S  Nelson       2.5
3067.0
14  1636000.0  Abbotsford      2    h      S  Nelson       2.5
3067.0

    Bedroom  Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
2         2         1    0     156.0          79.0       1900
```

```
4           3           2     0      134.0             150.0         1900
6           3           1     2      120.0             142.0         2014
11          4           2     0      245.0             210.0         1910
14          2           1     2      256.0             107.0         1890

           CouncilArea  Lattitude  Longtitude
Regionname  \
2   Yarra City Council   -37.8079    144.9934  Northern Metropolitan

4   Yarra City Council   -37.8093    144.9944  Northern Metropolitan

6   Yarra City Council   -37.8072    144.9941  Northern Metropolitan

11  Yarra City Council   -37.8024    144.9993  Northern Metropolitan

14  Yarra City Council   -37.8060    144.9954  Northern Metropolitan


    Propertycount
2          4019
4          4019
6          4019
11         4019
14         4019
```

Step 3. Data Exploration and analysis

```
from sklearn.model_selection import train_test_split
y = housing_pd["Price"]
X = housing_pd.drop(["Price"],axis=1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)

train_data = X_train.join(y_train)
# Remove duplicate columns
train_data = train_data.loc[:, ~train_data.columns.duplicated()]
train_data
```

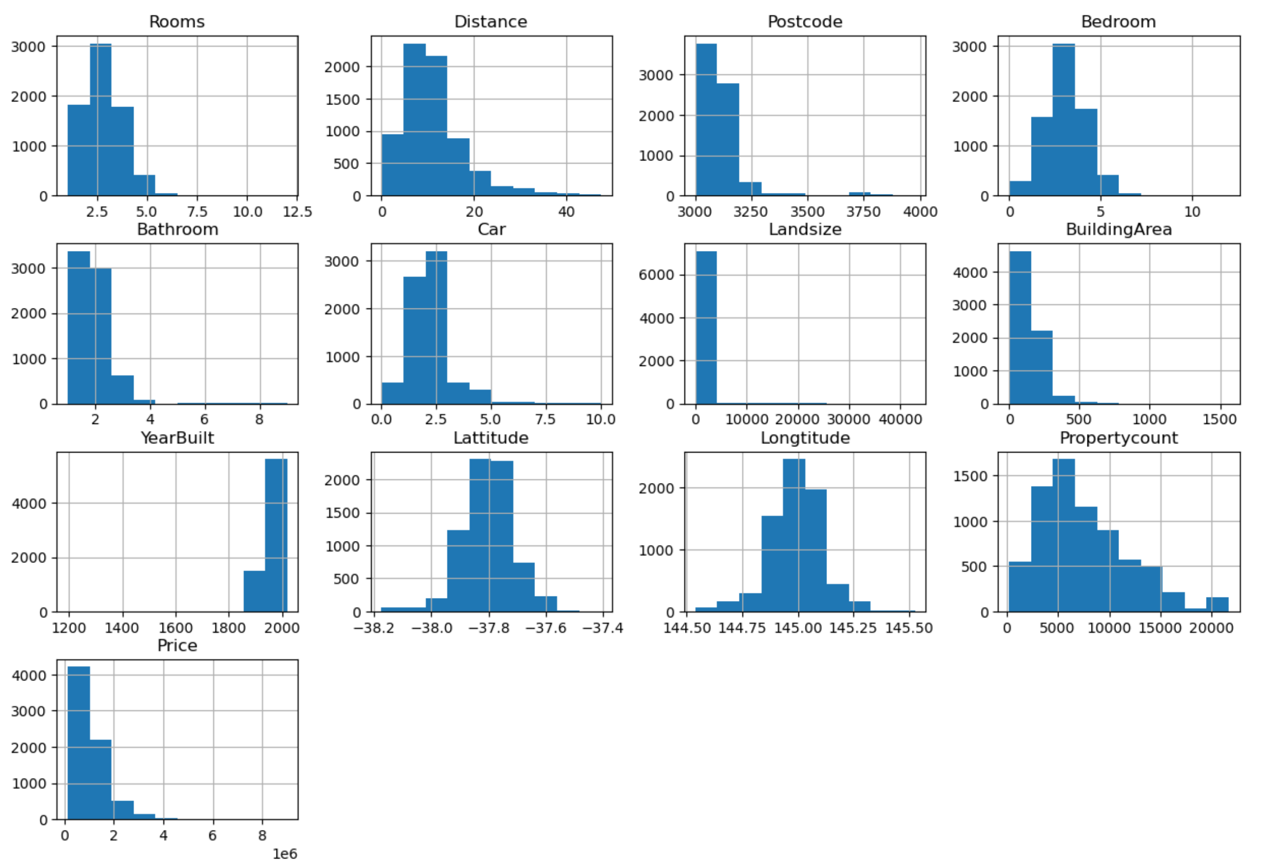
```
              Suburb  Rooms Type Method     Seller  Distance
Postcode  \
21848         Eltham      3    h     VB   Morrison      18.0
3095.0
178      Albert Park      4    h      S   Marshall       3.3
3206.0
34037      Brunswick      2    h     SP     Jellis       5.2
3056.0
30593      Mill Park      4    h      S        HAR      17.9
3082.0
3662       Doncaster      4    t      S  Fletchers      13.9
3108.0
```

```
...               ...    ...  ...    ...        ...       ...        ..
.
11091  Brunswick East      3    t      S     Nelson       4.5
3057.0
100      Airport West      3    h      S      Barry      13.5
3042.0
1059     Balwyn North      4    h     PI         RW       9.2
3104.0
19693   Sunshine West      3    h      S      Barry      10.5
3020.0
34478     Melton West      3    h      S   Reliance      31.7
3337.0

       Bedroom  Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
21848        3         2    2     868.0         135.0       1980
178          4         2    1     330.0         207.0       1910
34037        2         1    2     398.0         107.0       1890
30593        3         2    1     620.0         135.0       1980
3662         4         2    2     182.0         160.0       1998
...        ...       ...  ...       ...           ...        ...
11091        3         2    1       0.0         133.0       2014
100          3         1    2     971.0         113.0       1960
1059         4         3    2    1274.0         275.0       1970
19693        3         1    0     694.0         113.6       1950
34478        3         1    2     665.0         102.0       1975

                      CouncilArea  Lattitude  Longtitude  \
21848        Banyule City Council  -37.70659   145.16345
178     Port Phillip City Council  -37.84770   144.95580
34037       Moreland City Council  -37.76032   144.95981
30593     Whittlesea City Council  -37.65807   145.06132
3662      Manningham City Council  -37.78880   145.13800
...                           ...        ...         ...
11091       Moreland City Council  -37.77400   144.97310
100     Moonee Valley City Council  -37.71860   144.87600
1059      Boroondara City Council  -37.78220   145.09070
19693       Brimbank City Council  -37.78871   144.81369
34478         Melton City Council  -37.67239   144.56160

                  Regionname  Propertycount      Price
21848   Eastern Metropolitan           6990   810000.0
178    Southern Metropolitan           3280  4735000.0
34037  Northern Metropolitan          11918  1150000.0
30593  Northern Metropolitan          10529   660000.0
3662    Eastern Metropolitan           9028   985000.0
...                      ...            ...        ...
11091  Northern Metropolitan           5533  1010000.0
100     Western Metropolitan           3464   830000.0
1059   Southern Metropolitan           7809  2130000.0
19693   Western Metropolitan           6763   695000.0
```

```
34478       Northern Victoria            6065   371000.0

[7109 rows x 19 columns]
```

Histogram of the individual features

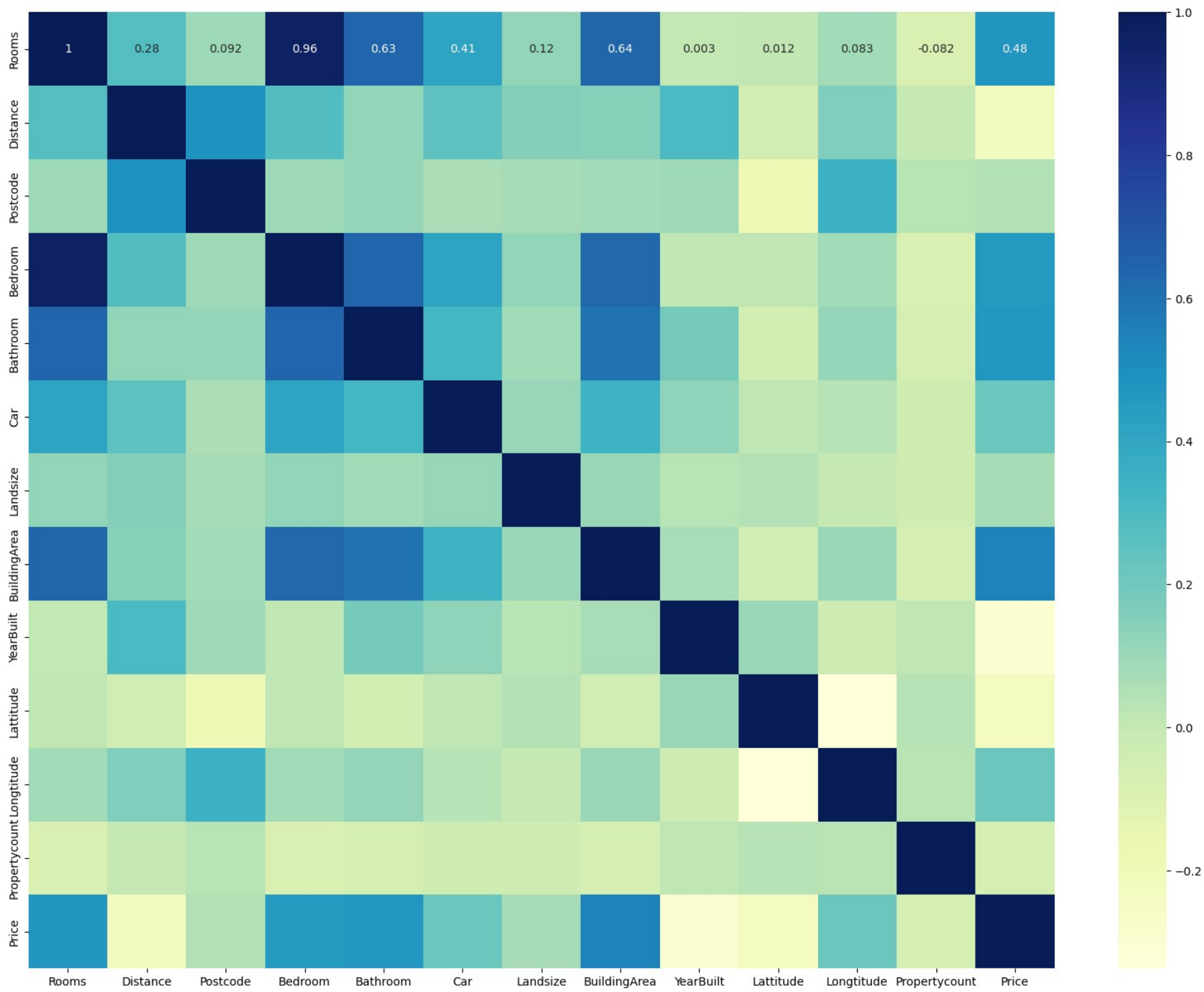
```
train_data.hist(figsize=(15,10))

array([[<Axes: title={'center': 'Rooms'}>,
        <Axes: title={'center': 'Distance'}>,
        <Axes: title={'center': 'Postcode'}>,
        <Axes: title={'center': 'Bedroom'}>],
       [<Axes: title={'center': 'Bathroom'}>,
        <Axes: title={'center': 'Car'}>,
        <Axes: title={'center': 'Landsize'}>,
        <Axes: title={'center': 'BuildingArea'}>],
       [<Axes: title={'center': 'YearBuilt'}>,
        <Axes: title={'center': 'Lattitude'}>,
        <Axes: title={'center': 'Longtitude'}>,
        <Axes: title={'center': 'Propertycount'}>],
       [<Axes: title={'center': 'Price'}>, <Axes: >, <Axes: >, <Axes:
>]],
      dtype=object)
```



Plotting the heatmap for the individual features

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(20,15))
sns.heatmap(train_data.drop(['Regionname',
'Suburb','Seller','CouncilArea','Type','Method'],axis=1).corr(),annot=
True,cmap='YlGnBu')
```

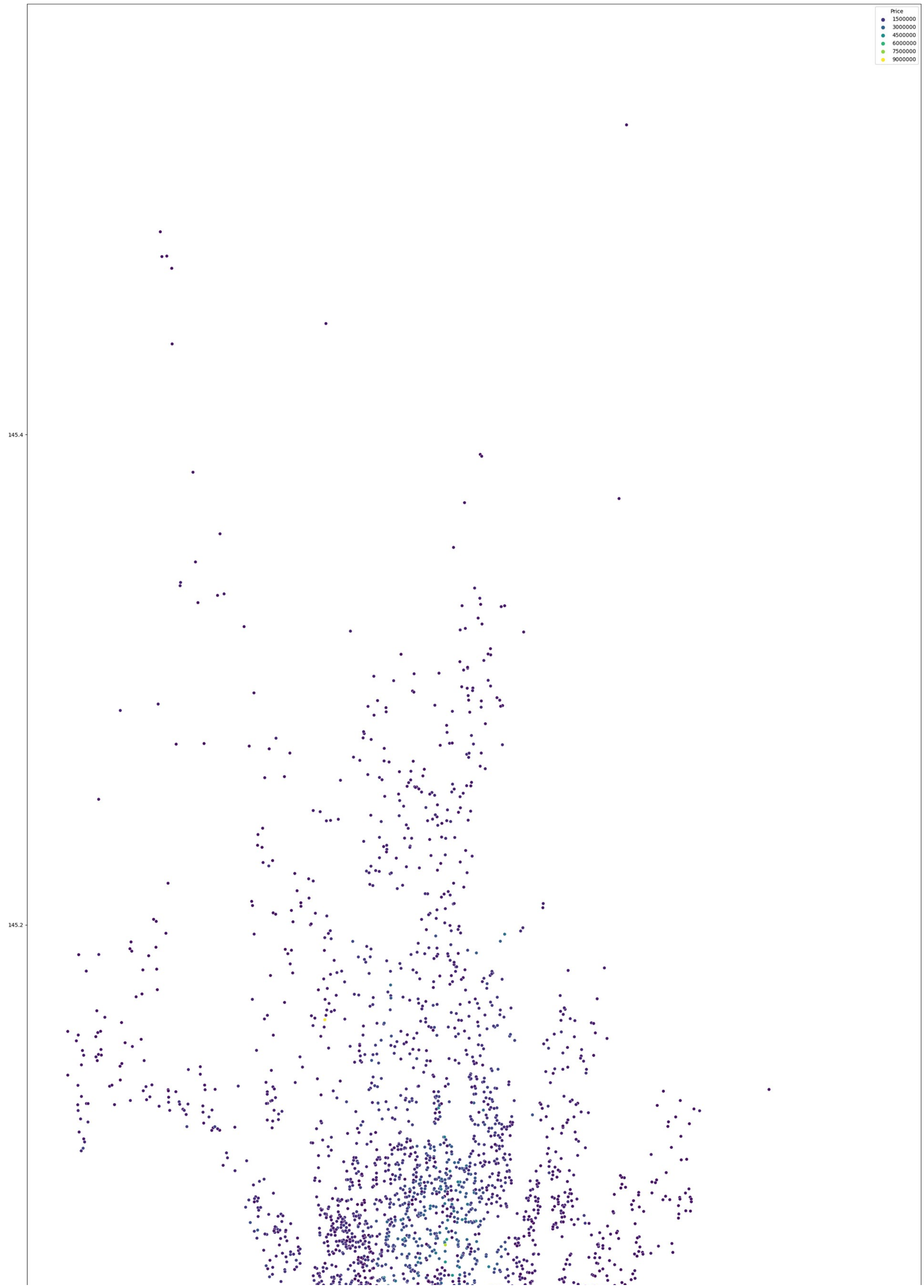
```
<Axes: >
```



Scatter plot of all the houses in Melbourne based on Price

```
plt.figure(figsize=(30,90))
sns.scatterplot(x = "Lattitude", y =
"Longtitude" ,data=train_data,hue="Price",palette = "viridis" )
```

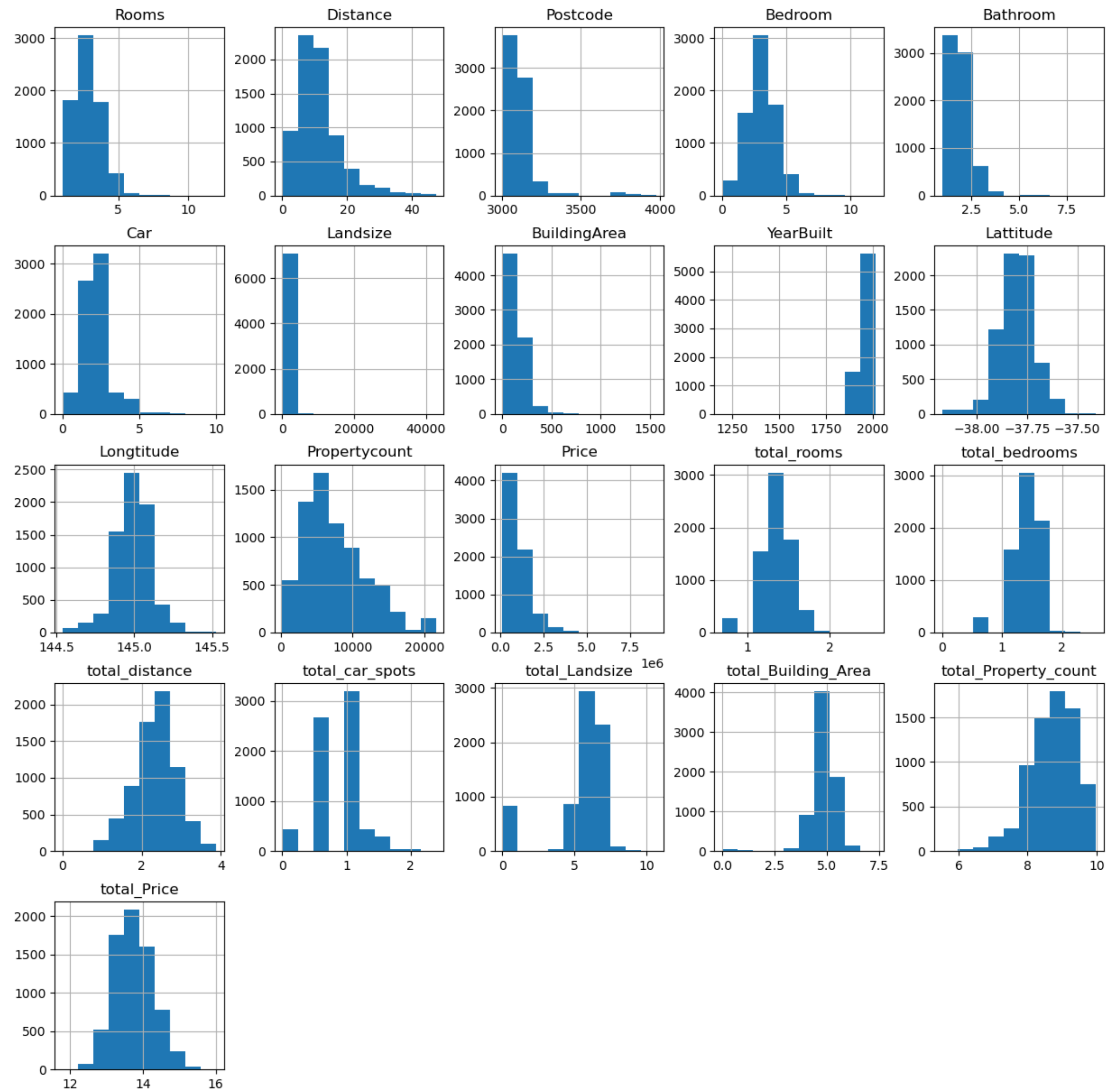
```
<Axes: xlabel='Lattitude', ylabel='Longtitude'>
```


Price
1500000
3000000
4500000
6000000
7500000
9000000
145.4
145.2

```python
# We add 1 inside for each log function to ensure our numeric values are valid
train_data['total_rooms'] = np.log(train_data['Rooms'] + 1)
train_data['total_bedrooms'] = np.log(train_data['Bedroom'] + 1)
train_data['total_distance'] = np.log(train_data['Distance'] + 1)
train_data['total_car_spots'] = np.log(train_data['Car'] + 1)
train_data['total_Landsize'] = np.log(train_data['Landsize'] + 1)
train_data['total_Building_Area'] = np.log(train_data['BuildingArea']
+ 1)
train_data['total_Property_count'] =
np.log(train_data['Propertycount'] + 1)
train_data['total_Price'] = np.log(train_data['Price'] + 1)
```

```python
train_data.hist(figsize=(15,15))
```

```
array([[<Axes: title={'center': 'Rooms'}>,
        <Axes: title={'center': 'Distance'}>,
        <Axes: title={'center': 'Postcode'}>,
        <Axes: title={'center': 'Bedroom'}>,
        <Axes: title={'center': 'Bathroom'}>],
       [<Axes: title={'center': 'Car'}>,
        <Axes: title={'center': 'Landsize'}>,
        <Axes: title={'center': 'BuildingArea'}>,
        <Axes: title={'center': 'YearBuilt'}>,
        <Axes: title={'center': 'Lattitude'}>],
       [<Axes: title={'center': 'Longtitude'}>,
        <Axes: title={'center': 'Propertycount'}>,
        <Axes: title={'center': 'Price'}>,
        <Axes: title={'center': 'total_rooms'}>,
        <Axes: title={'center': 'total_bedrooms'}>],
       [<Axes: title={'center': 'total_distance'}>,
        <Axes: title={'center': 'total_car_spots'}>,
        <Axes: title={'center': 'total_Landsize'}>,
        <Axes: title={'center': 'total_Building_Area'}>,
        <Axes: title={'center': 'total_Property_count'}>],
       [<Axes: title={'center': 'total_Price'}>, <Axes: >, <Axes: >,
        <Axes: >, <Axes: >]], dtype=object)
```



Step 4. Finalise the current data-frame

```
housing_pd['Regionname'].value_counts()

Regionname
Southern Metropolitan         2707
Northern Metropolitan         2612
Western Metropolitan          2059
Eastern Metropolitan           982
South-Eastern Metropolitan     371
Northern Victoria               62
```

```
Eastern Victoria                  51
Western Victoria                  43
Name: count, dtype: int64
```

```
housing_pd_shuffled = housing_pd.sample(n=len(housing_pd),random_state
= 1)
housing_pd_shuffled
```

```
           Price          Suburb  Rooms Type Method     Seller
Distance  \
17359   740000.0      Thomastown      3    h      S  Harcourts
15.3
17097   572000.0           Lalor      3    h      S       Love
16.3
5265   3225000.0        Hawthorn      4    h      S     Jellis
4.6
21286   626000.0          Jacana      3    h      S      Raine
14.0
9450    850000.0       Spotswood      4    h     VB         RT
7.7
...          ...             ...    ...  ...    ...        ...       .
..
8784    512000.0        Richmond      2    u      S   Marshall
2.6
29972   420000.0        Werribee      3    h      S      Barry
14.7
2848   3550000.0      Canterbury      5    h     VB         RT
9.0
16744   460000.0  Brunswick West      2    u      S        Ray
5.2
674    1950000.0       Ashburton      4    h      S        Tim
11.0

       Postcode  Bedroom  Bathroom  Car  Landsize  BuildingArea
YearBuilt  \
17359    3074.0        3         1    2     727.0         109.0
1952
17097    3075.0        3         1    2     640.0         140.0
1975
5265     3122.0        4         2    2     665.0         220.0
1890
21286    3047.0        3         1    2     622.0          87.0
1960
9450     3015.0        4         2    0     389.0         158.0
1990
...         ...      ...       ...  ...       ...           ...
...
8784     3121.0        2         1    1       0.0          61.0
1970
29972    3030.0        3         2    2     341.0         106.0
```

```
1995
2848     3126.0        5         4    4     684.0         427.0
2013
16744    3055.0        2         1    1      54.0          60.0
1970
674      3147.0        4         2    2     844.0         278.0
1940

                    CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan
...                         ...        ...         ...
...
8784         Yarra City Council  -37.81980   144.99600  Northern
Metropolitan
29972      Wyndham City Council  -37.88098   144.65754   Western
Metropolitan
2848    Boroondara City Council  -37.83200   145.08530  Southern
Metropolitan
16744     Moreland City Council  -37.75948   144.94758  Northern
Metropolitan
674     Boroondara City Council  -37.87150   145.06880  Southern
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
...              ...
8784           14949
29972          16166
2848            3265
16744           7082
674             3052

[8887 rows x 19 columns]
```

Catagorize the Region Name into categorical data (given numeric values)

```
pd.get_dummies(housing_pd_shuffled['Regionname'], dtype=float).head()
```

```
       Eastern Metropolitan  Eastern Victoria  Northern
Metropolitan  \
17359                   0.0               0.0                      1.0

17097                   0.0               0.0                      1.0

5265                    0.0               0.0                      0.0

21286                   0.0               0.0                      1.0

9450                    0.0               0.0                      0.0


       Northern Victoria  South-Eastern Metropolitan  Southern
Metropolitan  \
17359                0.0                         0.0
0.0
17097                0.0                         0.0
0.0
5265                 0.0                         0.0
1.0
21286                0.0                         0.0
0.0
9450                 0.0                         0.0
0.0

       Western Metropolitan  Western Victoria
17359                   0.0               0.0
17097                   0.0               0.0
5265                    0.0               0.0
21286                   0.0               0.0
9450                    1.0               0.0
```

We do the same for Council Area, Suburb, Type, Method, Seller, Type, Postcode and Method

```
pd.get_dummies(housing_pd_shuffled['CouncilArea'], dtype=float).head()
```

```
       Banyule City Council  Bayside City Council  Boroondara City
Council  \
17359                   0.0                   0.0
0.0
17097                   0.0                   0.0
0.0
```

```
5265                    0.0                     0.0
1.0
21286                   0.0                     0.0
0.0
9450                    0.0                     0.0
0.0

       Brimbank City Council  Cardinia Shire Council  Casey City
Council  \
17359                    0.0                     0.0
0.0
17097                    0.0                     0.0
0.0
5265                     0.0                     0.0
0.0
21286                    0.0                     0.0
0.0
9450                     0.0                     0.0
0.0

       Darebin City Council  Frankston City Council  Glen Eira City
Council  \
17359                   0.0                     0.0
0.0
17097                   0.0                     0.0
0.0
5265                    0.0                     0.0
0.0
21286                   0.0                     0.0
0.0
9450                    0.0                     0.0
0.0

       Greater Dandenong City Council  ...  Moorabool Shire Council  \
17359                             0.0  ...                      0.0
17097                             0.0  ...                      0.0
5265                              0.0  ...                      0.0
21286                             0.0  ...                      0.0
9450                              0.0  ...                      0.0

       Moreland City Council  Nillumbik Shire Council  \
17359                    0.0                      0.0
17097                    0.0                      0.0
5265                     0.0                      0.0
21286                    0.0                      0.0
9450                     0.0                      0.0

       Port Phillip City Council  Stonnington City Council  \
17359                        0.0                       0.0
17097                        0.0                       0.0
```

```
5265                           0.0                        0.0
21286                          0.0                        0.0
9450                           0.0                        0.0

       Whitehorse City Council  Whittlesea City Council  Wyndham City
Council  \
17359                      0.0                      1.0
0.0
17097                      0.0                      1.0
0.0
5265                       0.0                      0.0
0.0
21286                      0.0                      0.0
0.0
9450                       0.0                      0.0
0.0

       Yarra City Council  Yarra Ranges Shire Council
17359                 0.0                         0.0
17097                 0.0                         0.0
5265                  0.0                         0.0
21286                 0.0                         0.0
9450                  0.0                         0.0

[5 rows x 33 columns]
```

```
pd.get_dummies(housing_pd_shuffled['Suburb'], dtype=float).head()
```

```
       Abbotsford  Aberfeldie  Airport West  Albanvale  Albert Park
Albion  \
17359         0.0         0.0           0.0        0.0          0.0
0.0
17097         0.0         0.0           0.0        0.0          0.0
0.0
5265          0.0         0.0           0.0        0.0          0.0
0.0
21286         0.0         0.0           0.0        0.0          0.0
0.0
9450          0.0         0.0           0.0        0.0          0.0
0.0

       Alphington  Altona  Altona Meadows  Altona North  ...
Whittlesea  \
17359         0.0     0.0             0.0           0.0  ...
0.0
17097         0.0     0.0             0.0           0.0  ...
0.0
5265          0.0     0.0             0.0           0.0  ...
0.0
21286         0.0     0.0             0.0           0.0  ...
```

```
0.0
9450          0.0     0.0             0.0           0.0  ...
0.0

       Williams Landing  Williamstown  Williamstown North  Windsor
Wollert  \
17359               0.0           0.0                 0.0      0.0
0.0
17097               0.0           0.0                 0.0      0.0
0.0
5265                0.0           0.0                 0.0      0.0
0.0
21286               0.0           0.0                 0.0      0.0
0.0
9450                0.0           0.0                 0.0      0.0
0.0

       Wyndham Vale  Yallambie  Yarra Glen  Yarraville
17359           0.0        0.0         0.0         0.0
17097           0.0        0.0         0.0         0.0
5265            0.0        0.0         0.0         0.0
21286           0.0        0.0         0.0         0.0
9450            0.0        0.0         0.0         0.0

[5 rows x 315 columns]
```

```
pd.get_dummies(housing_pd_shuffled['Seller'], dtype=float).head()
```

```
       @Realty  Abercromby's  Ace  Alexkarbon  Allens  Anderson
Appleby  \
17359      0.0           0.0  0.0         0.0     0.0       0.0
0.0
17097      0.0           0.0  0.0         0.0     0.0       0.0
0.0
5265       0.0           0.0  0.0         0.0     0.0       0.0
0.0
21286      0.0           0.0  0.0         0.0     0.0       0.0
0.0
9450       0.0           0.0  0.0         0.0     0.0       0.0
0.0

       Aquire  Area  Ascend  ...  buyMyplace  hockingstuart  \
17359     0.0   0.0     0.0  ...         0.0            0.0
17097     0.0   0.0     0.0  ...         0.0            0.0
5265      0.0   0.0     0.0  ...         0.0            0.0
21286     0.0   0.0     0.0  ...         0.0            0.0
9450      0.0   0.0     0.0  ...         0.0            0.0

       hockingstuart/Advantage  hockingstuart/Biggin
hockingstuart/Village  \
```

```
17359                          0.0                        0.0
0.0
17097                          0.0                        0.0
0.0
5265                           0.0                        0.0
0.0
21286                          0.0                        0.0
0.0
9450                           0.0                        0.0
0.0

       hockingstuart/hockingstuart  iOne  iProperty  iSell  iTRAK
17359                          0.0   0.0        0.0    0.0    0.0
17097                          0.0   0.0        0.0    0.0    0.0
5265                           0.0   0.0        0.0    0.0    0.0
21286                          0.0   0.0        0.0    0.0    0.0
9450                           0.0   0.0        0.0    0.0    0.0

[5 rows x 250 columns]
```

```
pd.get_dummies(housing_pd_shuffled['Type'], dtype=float).head()
```

```
         h    t    u
17359  1.0  0.0  0.0
17097  1.0  0.0  0.0
5265   1.0  0.0  0.0
21286  1.0  0.0  0.0
9450   1.0  0.0  0.0
```

```
pd.get_dummies(housing_pd_shuffled['Method'], dtype=float).head()
```

```
        PI    S   SA   SP   VB
17359  0.0  1.0  0.0  0.0  0.0
17097  0.0  1.0  0.0  0.0  0.0
5265   0.0  1.0  0.0  0.0  0.0
21286  0.0  1.0  0.0  0.0  0.0
9450   0.0  0.0  0.0  0.0  1.0
```

```
pd.get_dummies(housing_pd_shuffled['Postcode'], dtype=int).head()
```

```
       3000.0  3002.0  3003.0  3006.0  3011.0  3012.0  3013.0  3015.0
3016.0  \
17359       0       0       0       0       0       0       0       0
0
17097       0       0       0       0       0       0       0       0
0
5265        0       0       0       0       0       0       0       0
0
21286       0       0       0       0       0       0       0       0
0
9450        0       0       0       0       0       0       0       1
```

```
0

       3018.0  ...  3803.0  3805.0  3806.0  3807.0  3808.0  3809.0
3810.0  \
17359       0  ...       0       0       0       0       0       0
0
17097       0  ...       0       0       0       0       0       0
0
5265        0  ...       0       0       0       0       0       0
0
21286       0  ...       0       0       0       0       0       0
0
9450        0  ...       0       0       0       0       0       0
0

       3910.0  3976.0  3977.0
17359       0       0       0
17097       0       0       0
5265        0       0       0
21286       0       0       0
9450        0       0       0

[5 rows x 194 columns]
```

We then drop the columns for Region Name, Council Area, Seller, Suburb and PostCode

```
housing_pd_shuffled.drop('Regionname',axis=1).head()
```

```
           Price      Suburb  Rooms Type Method     Seller
Distance  \
17359   740000.0  Thomastown      3    h      S  Harcourts      15.3

17097   572000.0       Lalor      3    h      S       Love      16.3

5265   3225000.0    Hawthorn      4    h      S     Jellis       4.6

21286   626000.0      Jacana      3    h      S      Raine      14.0

9450    850000.0   Spotswood      4    h     VB         RT       7.7


       Postcode  Bedroom  Bathroom  Car  Landsize  BuildingArea
YearBuilt  \
17359    3074.0        3         1    2     727.0         109.0
1952
17097    3075.0        3         1    2     640.0         140.0
1975
5265     3122.0        4         2    2     665.0         220.0
```

```
1890
21286    3047.0         3          1    2     622.0          87.0
1960
9450     3015.0         4          2    0     389.0         158.0
1990

                     CouncilArea  Lattitude  Longtitude  Propertycount

17359   Whittlesea City Council  -37.67873   145.01878           7955

17097   Whittlesea City Council  -37.65910   145.00549           8279

5265    Boroondara City Council  -37.81400   145.01750          11308

21286         Hume City Council  -37.68908   144.91160            851

9450   Hobsons Bay City Council  -37.82840   144.88610           1223
```

```
housing_pd_shuffled.drop('CouncilArea',axis=1).head()
```

```
           Price     Suburb  Rooms Type Method     Seller
Distance  \
17359   740000.0  Thomastown      3    h      S  Harcourts      15.3

17097   572000.0       Lalor      3    h      S       Love      16.3

5265   3225000.0    Hawthorn      4    h      S     Jellis       4.6

21286   626000.0      Jacana      3    h      S      Raine      14.0

9450    850000.0   Spotswood      4    h     VB         RT       7.7


       Postcode  Bedroom  Bathroom  Car  Landsize  BuildingArea
YearBuilt  \
17359    3074.0         3          1    2     727.0         109.0
1952
17097    3075.0         3          1    2     640.0         140.0
1975
5265     3122.0         4          2    2     665.0         220.0
1890
21286    3047.0         3          1    2     622.0          87.0
1960
9450     3015.0         4          2    0     389.0         158.0
1990

       Lattitude  Longtitude             Regionname  Propertycount
17359  -37.67873   145.01878  Northern Metropolitan           7955
17097  -37.65910   145.00549  Northern Metropolitan           8279
5265   -37.81400   145.01750  Southern Metropolitan          11308
```

```
21286  -37.68908   144.91160  Northern Metropolitan            851
9450   -37.82840   144.88610   Western Metropolitan           1223
```

```
housing_pd_shuffled.drop('Suburb',axis=1).head()
```

```
           Price  Rooms Type Method     Seller  Distance  Postcode
Bedroom  \
17359   740000.0      3    h      S  Harcourts      15.3    3074.0
3
17097   572000.0      3    h      S       Love      16.3    3075.0
3
5265   3225000.0      4    h      S     Jellis       4.6    3122.0
4
21286   626000.0      3    h      S      Raine      14.0    3047.0
3
9450    850000.0      4    h     VB         RT       7.7    3015.0
4

       Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
17359         1    2     727.0         109.0       1952
17097         1    2     640.0         140.0       1975
5265          2    2     665.0         220.0       1890
21286         1    2     622.0          87.0       1960
9450          2    0     389.0         158.0       1990

                   CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
```

```
housing_pd_shuffled.drop('Seller',axis=1).head()
```

```
           Price      Suburb  Rooms Type Method  Distance  Postcode
Bedroom  \
17359   740000.0  Thomastown      3    h      S      15.3    3074.0
```

```
3
17097   572000.0       Lalor       3    h       S       16.3    3075.0
3
5265   3225000.0    Hawthorn       4    h       S        4.6    3122.0
4
21286   626000.0      Jacana       3    h       S       14.0    3047.0
3
9450    850000.0   Spotswood       4    h      VB        7.7    3015.0
4

       Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
17359         1    2     727.0         109.0       1952
17097         1    2     640.0         140.0       1975
5265          2    2     665.0         220.0       1890
21286         1    2     622.0          87.0       1960
9450          2    0     389.0         158.0       1990

                    CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
```

```
housing_pd_shuffled.drop('Type',axis=1).head()
```

```
           Price      Suburb  Rooms Method     Seller  Distance
Postcode  \
17359   740000.0  Thomastown      3      S  Harcourts      15.3
3074.0
17097   572000.0       Lalor      3      S       Love      16.3
3075.0
5265   3225000.0    Hawthorn      4      S     Jellis       4.6
3122.0
21286   626000.0      Jacana      3      S      Raine      14.0
3047.0
9450    850000.0   Spotswood      4     VB         RT       7.7
```

```
3015.0

       Bedroom  Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
17359        3         1    2     727.0         109.0       1952
17097        3         1    2     640.0         140.0       1975
5265         4         2    2     665.0         220.0       1890
21286        3         1    2     622.0          87.0       1960
9450         4         2    0     389.0         158.0       1990

                    CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
```

```
housing_pd_shuffled.drop('Method',axis=1).head()
```

```
           Price      Suburb  Rooms Type     Seller  Distance
Postcode  \
17359   740000.0  Thomastown      3    h  Harcourts      15.3
3074.0
17097   572000.0       Lalor      3    h       Love      16.3
3075.0
5265   3225000.0    Hawthorn      4    h     Jellis       4.6
3122.0
21286   626000.0      Jacana      3    h      Raine      14.0
3047.0
9450    850000.0   Spotswood      4    h         RT       7.7
3015.0

       Bedroom  Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
17359        3         1    2     727.0         109.0       1952
17097        3         1    2     640.0         140.0       1975
5265         4         2    2     665.0         220.0       1890
21286        3         1    2     622.0          87.0       1960
9450         4         2    0     389.0         158.0       1990
```

```
                        CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
```

```
housing_pd_shuffled.drop('Postcode',axis=1).head()
```

```
             Price      Suburb  Rooms Type Method     Seller  Distance
Bedroom  \
17359   740000.0  Thomastown      3    h      S  Harcourts      15.3
3
17097   572000.0       Lalor      3    h      S       Love      16.3
3
5265   3225000.0    Hawthorn      4    h      S     Jellis       4.6
4
21286   626000.0      Jacana      3    h      S      Raine      14.0
3
9450    850000.0   Spotswood      4    h     VB         RT       7.7
4

       Bathroom  Car  Landsize  BuildingArea  YearBuilt  \
17359         1    2     727.0         109.0       1952
17097         1    2     640.0         140.0       1975
5265          2    2     665.0         220.0       1890
21286         1    2     622.0          87.0       1960
9450          2    0     389.0         158.0       1990

                        CouncilArea  Lattitude  Longtitude
Regionname  \
17359   Whittlesea City Council  -37.67873   145.01878  Northern
Metropolitan
17097   Whittlesea City Council  -37.65910   145.00549  Northern
Metropolitan
5265    Boroondara City Council  -37.81400   145.01750  Southern
```

```
Metropolitan
21286         Hume City Council  -37.68908   144.91160  Northern
Metropolitan
9450   Hobsons Bay City Council  -37.82840   144.88610   Western
Metropolitan

       Propertycount
17359           7955
17097           8279
5265           11308
21286            851
9450            1223
```

Here we are creating our final dataframe for the house in Melbourne dataset

```
housing_pd_rn =
pd.concat([housing_pd_shuffled.drop('Regionname',axis=1),pd.get_dummie
s(housing_pd_shuffled['Regionname'],dtype=float)], axis=1)
housing_pd_ca =
pd.concat([housing_pd_shuffled.drop('CouncilArea',axis=1),pd.get_dummi
es(housing_pd_shuffled['CouncilArea'],dtype=float)], axis=1)
housing_pd_s =
pd.concat([housing_pd_shuffled.drop('Suburb',axis=1),pd.get_dummies(ho
using_pd_shuffled['Suburb'],dtype=float)], axis=1)
housing_pd_sg =
pd.concat([housing_pd_shuffled.drop('Seller',axis=1),pd.get_dummies(ho
using_pd_shuffled['Seller'],dtype=float)], axis=1)
housing_pd_t =
pd.concat([housing_pd_shuffled.drop('Type',axis=1),pd.get_dummies(hous
ing_pd_shuffled['Type'],dtype=float)], axis=1)
housing_pd_m =
pd.concat([housing_pd_shuffled.drop('Method',axis=1),pd.get_dummies(ho
using_pd_shuffled['Method'],dtype=float)], axis=1)
housing_pd_p =
pd.concat([housing_pd_shuffled.drop('Postcode',axis=1),pd.get_dummies(
housing_pd_shuffled['Postcode'],dtype=float)], axis=1)
housing_pd_rn_ca = pd.concat([housing_pd_rn,housing_pd_ca], axis=1)
housing_pd_s_sg = pd.concat([housing_pd_s,housing_pd_sg], axis=1)
housing_pd_t_m = pd.concat([housing_pd_t,housing_pd_m], axis=1)
housing_pd_t_m_p = pd.concat([housing_pd_t_m,housing_pd_p], axis=1)

housing_pd_final_rn_ca_s_sg =
pd.concat([housing_pd_rn_ca ,housing_pd_s_sg], axis=1)
housing_pd_final =
pd.concat([housing_pd_final_rn_ca_s_sg ,housing_pd_t_m_p], axis=1)
housing_pd_final =
housing_pd_final.drop(['Regionname','CouncilArea','Suburb','Type','Met
```

```
hod','Seller','Postcode'],axis=1)
housing_pd_final.head()
```

```
           Price  Rooms  Distance  Bedroom  Bathroom  Car  Landsize  \
17359   740000.0      3      15.3        3         1    2     727.0
17097   572000.0      3      16.3        3         1    2     640.0
5265   3225000.0      4       4.6        4         2    2     665.0
21286   626000.0      3      14.0        3         1    2     622.0
9450    850000.0      4       7.7        4         2    0     389.0

       BuildingArea  YearBuilt  Lattitude  ...  3803.0  3805.0  3806.0
\
17359         109.0       1952  -37.67873  ...     0.0     0.0     0.0

17097         140.0       1975  -37.65910  ...     0.0     0.0     0.0

5265          220.0       1890  -37.81400  ...     0.0     0.0     0.0

21286          87.0       1960  -37.68908  ...     0.0     0.0     0.0

9450          158.0       1990  -37.82840  ...     0.0     0.0     0.0


       3807.0  3808.0  3809.0  3810.0  3910.0  3976.0  3977.0
17359     0.0     0.0     0.0     0.0     0.0     0.0     0.0
17097     0.0     0.0     0.0     0.0     0.0     0.0     0.0
5265      0.0     0.0     0.0     0.0     0.0     0.0     0.0
21286     0.0     0.0     0.0     0.0     0.0     0.0     0.0
9450      0.0     0.0     0.0     0.0     0.0     0.0     0.0

[5 rows x 892 columns]
```

Checking the shape and types of the final Melbourne housing data-frame

```
housing_pd_final.dtypes
```

```
Price        float64
Rooms          int64
Distance     float64
Bedroom        int32
Bathroom       int32
              ...
3809.0       float64
3810.0       float64
3910.0       float64
3976.0       float64
```

```
3977.0      float64
Length: 892, dtype: object
```

```
housing_pd_final.shape
```

```
(8887, 892)
```

```
# Remove duplicate columns
housing_pd_final = housing_pd_final.loc[:,
~housing_pd_final.columns.duplicated()]
housing_pd_final
```

```
           Price  Rooms  Distance  Bedroom  Bathroom  Car  Landsize  \
17359   740000.0      3      15.3        3         1    2     727.0
17097   572000.0      3      16.3        3         1    2     640.0
5265   3225000.0      4       4.6        4         2    2     665.0
21286   626000.0      3      14.0        3         1    2     622.0
9450    850000.0      4       7.7        4         2    0     389.0
...          ...    ...       ...      ...       ...  ...       ...
8784    512000.0      2       2.6        2         1    1       0.0
29972   420000.0      3      14.7        3         2    2     341.0
2848   3550000.0      5       9.0        5         4    4     684.0
16744   460000.0      2       5.2        2         1    1      54.0
674    1950000.0      4      11.0        4         2    2     844.0

       BuildingArea  YearBuilt  Lattitude  ...  3803.0  3805.0  3806.0
\
17359         109.0       1952  -37.67873  ...     0.0     0.0     0.0

17097         140.0       1975  -37.65910  ...     0.0     0.0     0.0

5265          220.0       1890  -37.81400  ...     0.0     0.0     0.0

21286          87.0       1960  -37.68908  ...     0.0     0.0     0.0

9450          158.0       1990  -37.82840  ...     0.0     0.0     0.0

...             ...        ...        ...  ...     ...     ...     ...

8784           61.0       1970  -37.81980  ...     0.0     0.0     0.0

29972         106.0       1995  -37.88098  ...     0.0     0.0     0.0

2848          427.0       2013  -37.83200  ...     0.0     0.0     0.0

16744          60.0       1970  -37.75948  ...     0.0     0.0     0.0

674           278.0       1940  -37.87150  ...     0.0     0.0     0.0


       3807.0  3808.0  3809.0  3810.0  3910.0  3976.0  3977.0
```

```
17359     0.0     0.0     0.0     0.0     0.0     0.0     0.0
17097     0.0     0.0     0.0     0.0     0.0     0.0     0.0
5265      0.0     0.0     0.0     0.0     0.0     0.0     0.0
21286     0.0     0.0     0.0     0.0     0.0     0.0     0.0
9450      0.0     0.0     0.0     0.0     0.0     0.0     0.0
...       ...     ...     ...     ...     ...     ...     ...
8784      0.0     0.0     0.0     0.0     0.0     0.0     0.0
29972     0.0     0.0     0.0     0.0     0.0     0.0     0.0
2848      0.0     0.0     0.0     0.0     0.0     0.0     0.0
16744     0.0     0.0     0.0     0.0     0.0     0.0     0.0
674       0.0     0.0     0.0     0.0     0.0     0.0     0.0

[8887 rows x 819 columns]
```

```
housing_pd_final.columns.values
```

```
array(['Price', 'Rooms', 'Distance', 'Bedroom', 'Bathroom', 'Car',
       'Landsize', 'BuildingArea', 'YearBuilt', 'Lattitude',
'Longtitude',
       'Propertycount', 'Eastern Metropolitan', 'Eastern Victoria',
       'Northern Metropolitan', 'Northern Victoria',
       'South-Eastern Metropolitan', 'Southern Metropolitan',
       'Western Metropolitan', 'Western Victoria', 'Banyule City
Council',
       'Bayside City Council', 'Boroondara City Council',
       'Brimbank City Council', 'Cardinia Shire Council',
       'Casey City Council', 'Darebin City Council',
       'Frankston City Council', 'Glen Eira City Council',
       'Greater Dandenong City Council', 'Hobsons Bay City Council',
       'Hume City Council', 'Kingston City Council', 'Knox City
Council',
       'Macedon Ranges Shire Council', 'Manningham City Council',
       'Maribyrnong City Council', 'Maroondah City Council',
       'Melbourne City Council', 'Melton City Council',
       'Mitchell Shire Council', 'Monash City Council',
       'Moonee Valley City Council', 'Moorabool Shire Council',
       'Moreland City Council', 'Nillumbik Shire Council',
       'Port Phillip City Council', 'Stonnington City Council',
       'Whitehorse City Council', 'Whittlesea City Council',
       'Wyndham City Council', 'Yarra City Council',
       'Yarra Ranges Shire Council', 'Abbotsford', 'Aberfeldie',
       'Airport West', 'Albanvale', 'Albert Park', 'Albion',
'Alphington',
       'Altona', 'Altona Meadows', 'Altona North', 'Ardeer',
'Armadale',
       'Ascot Vale', 'Ashburton', 'Ashwood', 'Aspendale',
       'Aspendale Gardens', 'Attwood', 'Avondale Heights',
       'Bacchus Marsh', 'Balaclava', 'Balwyn', 'Balwyn North',
       'Bayswater', 'Bayswater North', 'Beaconsfield',
       'Beaconsfield Upper', 'Beaumaris', 'Bellfield', 'Bentleigh',
```

```
        'Bentleigh East', 'Berwick', 'Black Rock', 'Blackburn',
        'Blackburn North', 'Blackburn South', 'Bonbeach', 'Boronia',
        'Botanic Ridge', 'Box Hill', 'Braybrook', 'Briar Hill',
'Brighton',
        'Brighton East', 'Broadmeadows', 'Brookfield', 'Brooklyn',
        'Brunswick', 'Brunswick East', 'Brunswick West', 'Bulleen',
        'Bullengarook', 'Bundoora', 'Burnley', 'Burnside',
        'Burnside Heights', 'Burwood', 'Burwood East', 'Cairnlea',
        'Camberwell', 'Campbellfield', 'Canterbury', 'Carlton',
        'Carlton North', 'Carnegie', 'Caroline Springs', 'Carrum',
        'Carrum Downs', 'Caulfield', 'Caulfield East', 'Caulfield
North',
        'Caulfield South', 'Chadstone', 'Chelsea', 'Chelsea Heights',
        'Cheltenham', 'Chirnside Park', 'Clarinda', 'Clayton',
        'Clayton South', 'Clifton Hill', 'Coburg', 'Coburg North',
        'Collingwood', 'Coolaroo', 'Craigieburn', 'Cranbourne',
        'Cranbourne North', 'Cremorne', 'Croydon', 'Croydon Hills',
        'Croydon North', 'Croydon South', 'Dallas', 'Dandenong',
        'Dandenong North', 'Deepdene', 'Deer Park', 'Delahey',
'Derrimut',
        'Diamond Creek', 'Diggers Rest', 'Dingley Village',
'Doncaster',
        'Doncaster East', 'Donvale', 'Doreen', 'Doveton', 'Eaglemont',
        'East Melbourne', 'Edithvale', 'Elsternwick', 'Eltham',
        'Eltham North', 'Elwood', 'Emerald', 'Endeavour Hills',
'Epping',
        'Essendon', 'Essendon North', 'Essendon West', 'Fairfield',
        'Fawkner', 'Ferntree Gully', 'Fitzroy', 'Fitzroy North',
        'Flemington', 'Footscray', 'Forest Hill', 'Frankston',
        'Frankston North', 'Frankston South', 'Gardenvale', 'Gisborne',
        'Gisborne South', 'Gladstone Park', 'Glen Huntly', 'Glen Iris',
        'Glen Waverley', 'Glenroy', 'Gowanbrae', 'Greensborough',
        'Greenvale', 'Hadfield', 'Hallam', 'Hampton', 'Hampton East',
        'Hampton Park', 'Hawthorn', 'Hawthorn East', 'Healesville',
        'Heathmont', 'Heidelberg', 'Heidelberg Heights', 'Heidelberg
West',
        'Highett', 'Hillside', 'Hoppers Crossing', 'Hughesdale',
        'Huntingdale', 'Hurstbridge', 'Ivanhoe', 'Ivanhoe East',
'Jacana',
        'Kealba', 'Keilor', 'Keilor Downs', 'Keilor East', 'Keilor
Lodge',
        'Keilor Park', 'Kensington', 'Kew', 'Kew East', 'Keysborough',
        'Kilsyth', 'Kings Park', 'Kingsbury', 'Kingsville',
'Knoxfield',
        'Kooyong', 'Kurunjang', 'Lalor', 'Langwarrin', 'Lower Plenty',
        'Lysterfield', 'Maidstone', 'Malvern', 'Malvern East',
        'Maribyrnong', 'McKinnon', 'Meadow Heights', 'Melbourne',
'Melton',
        'Melton South', 'Melton West', 'Mentone', 'Mernda',
```

```
'Mickleham',
       'Middle Park', 'Mill Park', 'Mitcham', 'Mont Albert',
       'Montmorency', 'Montrose', 'Moonee Ponds', 'Moorabbin',
       'Mooroolbark', 'Mordialloc', 'Mount Evelyn', 'Mount Waverley',
       'Mulgrave', 'Murrumbeena', 'Narre Warren', 'Newport',
'Niddrie',
       'Noble Park', 'North Melbourne', 'North Warrandyte',
'Northcote',
       'Notting Hill', 'Nunawading', 'Oak Park', 'Oakleigh',
       'Oakleigh East', 'Oakleigh South', 'Officer', 'Ormond',
'Pakenham',
       'Parkdale', 'Parkville', 'Pascoe Vale', 'Patterson Lakes',
       'Plumpton', 'Point Cook', 'Port Melbourne', 'Prahran',
'Preston',
       'Princes Hill', 'Research', 'Reservoir', 'Richmond',
       'Riddells Creek', 'Ringwood', 'Ringwood East', 'Ringwood
North',
       'Ripponlea', 'Rosanna', 'Rowville', 'Roxburgh Park',
'Sandhurst',
       'Sandringham', 'Scoresby', 'Seabrook', 'Seaford', 'Seaholme',
       'Seddon', 'Skye', 'South Kingsville', 'South Melbourne',
       'South Morang', 'South Yarra', 'Southbank', 'Spotswood',
       'Springvale', 'Springvale South', 'St Albans', 'St Helena',
       'St Kilda', 'Strathmore', 'Strathmore Heights', 'Sunbury',
       'Sunshine', 'Sunshine North', 'Sunshine West', 'Surrey Hills',
       'Sydenham', 'Tarneit', 'Taylors Hill', 'Taylors Lakes',
       'Templestowe', 'Templestowe Lower', 'The Basin', 'Thomastown',
       'Thornbury', 'Toorak', 'Travancore', 'Truganina',
'Tullamarine',
       'Upwey', 'Vermont', 'Vermont South', 'Viewbank', 'Wallan',
       'Wantirna', 'Wantirna South', 'Warrandyte', 'Waterways',
       'Watsonia', 'Watsonia North', 'Wattle Glen', 'Werribee',
       'West Footscray', 'West Melbourne', 'Westmeadows', 'Wheelers
Hill',
       'Whittlesea', 'Williams Landing', 'Williamstown',
       'Williamstown North', 'Windsor', 'Wollert', 'Wyndham Vale',
       'Yallambie', 'Yarra Glen', 'Yarraville', '@Realty',
"Abercromby's",
       'Ace', 'Alexkarbon', 'Allens', 'Anderson', 'Appleby', 'Aquire',
       'Area', 'Ascend', 'Ash', 'Assisi', 'Australian', 'Avion',
'Barlow',
       'Barry', 'Bayside', 'Bekdon', 'Beller', 'Bells', 'Benlor',
       'Besser', 'Biggin', 'Boran', 'Boutique', 'Bowman', 'Brace',
'Brad',
       'Buckingham', 'Bullen', 'Burnham', 'Burns', 'Buxton',
       'Buxton/Advantage', 'Buxton/Find', 'C21', 'Caine', 'Calder',
       'Carter', 'Castran', 'Cayzer', 'Century', 'Chambers',
'Charlton',
       'Charter', 'Chisholm', 'Christopher', 'Clairmont', 'Collings',
```

```
       'Collins', 'Commercial', 'Compton', 'Considine', 'Crane',
       "D'Aprano", 'Daniel', 'Darras', 'Darren', 'Del', 'Dingle',
'Dixon',
       'Domain', 'Douglas', 'Edward', 'Eric', 'Eview', 'FN', 'First',
       'Flannagan', 'Fletchers', 'Fletchers/One', 'Follett', 'Frank',
       'GL', 'Galldon', 'Gardiner', 'Garvey', 'Gary', 'Geoff',
'Grantham',
       'Greg', 'Gunn&Co', 'H', 'HAR', 'Hall', 'Harcourts',
'Harrington',
       'Haughton', 'Hayeswinckle', 'Hodges', 'Holland', 'Homes',
       'Hoskins', 'Hunter', 'Iconek', 'J', 'JMRE', 'JRW', 'Jas',
'Jason',
       'Jellis', 'Johnston', 'Joseph', 'Just', 'Justin', 'Kay',
'Kaye',
       'Kelly', 'Ken', 'L', 'LITTLE', 'LJ', 'LJH', 'LLC', 'Langwell',
       'Le', 'Leading', 'Leeburn', 'Leyton', 'Lindellas', 'Love',
'Lucas',
       'Luxton', 'MICM', 'Maddison', 'Maitland', 'Mandy', 'Marshall',
       'Mason', 'Matthew', 'Max', 'McDonald', 'McGrath',
'McGrath/First',
       'McGrath/Langwell', 'McLennan', 'McNaughton', 'Miles',
       'Millership', 'Mitchell', 'Moonee', 'Morleys', 'Morrison',
'Munn',
       'Naison', 'Nardella', 'Nelson', 'New', 'Nicholson', 'Nick',
'Noel',
       "O'Brien", "O'Donoghues", 'OBrien', 'Oak', 'Obrien', 'One',
'Only',
       'Owen', 'PRDNationwide', 'PSP', 'Pagan', 'Parkes', 'Paul',
'Peake',
       'Peter', 'Philip', 'Point', 'Pride', 'Prime',
"Private/Tiernan's",
       'Prof.', 'Property', 'Prowse', 'Purplebricks', 'R&H', 'RE',
       'REMAX', 'RT', 'RW', 'Raine', 'Raine&Horne', 'Ray', 'Re',
'Real',
       'Red', 'Redina', 'Reed', 'Reliance', 'Rendina', 'Rexhepi',
       'Ristic', 'Rodney', 'Ross', 'Rounds', 'Ryder', 'S&L',
'Sanctuary',
       'Schroeder', 'Scott', 'Sell', 'Skad', "Sotheby's", 'Stockdale',
       'Sweeney', 'Sweeney/Advantage', 'TRUE', 'The', 'Thomson',
       "Tiernan's", 'Tim', 'Trimson', 'Triwest', 'U', 'Upper',
'Upside',
       'VICPROP', 'VICProp', 'Veitch', 'Vic', 'Victory', 'Village',
       'W.B.', 'WHITEFOX', 'Walsh', 'Walshe', 'Weast', 'Weston',
       'Westside', 'Whiting', 'William', 'Williams', 'Wilson', 'Win',
       'Wood', 'Woodards', 'Wyndham', 'YPA', 'Zahn', 'buyMyplace',
       'hockingstuart', 'hockingstuart/Advantage',
'hockingstuart/Biggin',
       'hockingstuart/Village', 'hockingstuart/hockingstuart', 'iOne',
       'iProperty', 'iSell', 'iTRAK', 'h', 't', 'u', 'PI', 'S', 'SA',
```

```
       'SP', 'VB', 3000.0, 3002.0, 3003.0, 3006.0, 3011.0, 3012.0,
3013.0,
       3015.0, 3016.0, 3018.0, 3019.0, 3020.0, 3021.0, 3022.0, 3023.0,
       3024.0, 3025.0, 3027.0, 3028.0, 3029.0, 3030.0, 3031.0, 3032.0,
       3033.0, 3034.0, 3036.0, 3037.0, 3038.0, 3039.0, 3040.0, 3041.0,
       3042.0, 3043.0, 3044.0, 3046.0, 3047.0, 3048.0, 3049.0, 3051.0,
       3052.0, 3053.0, 3054.0, 3055.0, 3056.0, 3057.0, 3058.0, 3059.0,
       3060.0, 3061.0, 3064.0, 3065.0, 3066.0, 3067.0, 3068.0, 3070.0,
       3071.0, 3072.0, 3073.0, 3074.0, 3075.0, 3076.0, 3078.0, 3079.0,
       3081.0, 3082.0, 3083.0, 3084.0, 3085.0, 3087.0, 3088.0, 3089.0,
       3093.0, 3094.0, 3095.0, 3096.0, 3099.0, 3101.0, 3102.0, 3103.0,
       3104.0, 3105.0, 3106.0, 3107.0, 3108.0, 3109.0, 3111.0, 3113.0,
       3116.0, 3121.0, 3122.0, 3123.0, 3124.0, 3125.0, 3126.0, 3127.0,
       3128.0, 3130.0, 3131.0, 3132.0, 3133.0, 3134.0, 3135.0, 3136.0,
       3137.0, 3138.0, 3141.0, 3142.0, 3143.0, 3144.0, 3145.0, 3146.0,
       3147.0, 3148.0, 3149.0, 3150.0, 3151.0, 3152.0, 3153.0, 3154.0,
       3155.0, 3156.0, 3158.0, 3161.0, 3162.0, 3163.0, 3165.0, 3166.0,
       3167.0, 3168.0, 3169.0, 3170.0, 3171.0, 3172.0, 3173.0, 3174.0,
       3175.0, 3177.0, 3178.0, 3179.0, 3180.0, 3181.0, 3182.0, 3183.0,
       3184.0, 3185.0, 3186.0, 3187.0, 3188.0, 3189.0, 3190.0, 3191.0,
       3192.0, 3193.0, 3194.0, 3195.0, 3196.0, 3197.0, 3198.0, 3199.0,
       3200.0, 3201.0, 3204.0, 3205.0, 3206.0, 3207.0, 3335.0, 3337.0,
       3338.0, 3340.0, 3427.0, 3429.0, 3431.0, 3437.0, 3750.0, 3752.0,
       3754.0, 3756.0, 3757.0, 3765.0, 3775.0, 3777.0, 3782.0, 3796.0,
       3802.0, 3803.0, 3805.0, 3806.0, 3807.0, 3808.0, 3809.0, 3810.0,
       3910.0, 3976.0, 3977.0], dtype=object)
```

Step 5. Spliting the final housing dataframe for the training, testing and validation data.

```
# You can train, for demonstration purposes we use about 7000 to
train, the next 900 to test and the rest used for testing data
# These results will vary depending on the rows you imputed or cleaned
in Step 2.
train_pd,test_pd,val_pd =
housing_pd_final[:7000],housing_pd_final[7000:7900],housing_pd_final[7
900:]
len(train_pd), len(test_pd),len(val_pd)
```

```
(7000, 900, 987)
```

Compute our training data

```
X_train,y_train = train_pd.to_numpy()[:,1:], train_pd.to_numpy()[:,0]
y_train
```

```
array([ 740000.,  572000., 3225000., ...,  750000.,  750000.,
580000.])
```

```
y_train.shape
```

```
(7000,)
```

```
X_train.shape
```

```
(7000, 818)
```

Compute our testing and validation data

```
X_test,y_test= test_pd.to_numpy()[:,1:], test_pd.to_numpy()[:,0]
X_val,y_val = val_pd.to_numpy()[:,1:], val_pd.to_numpy()[:,0]
X_test.shape,y_test.shape,X_val.shape,y_val.shape
```

```
((900, 818), (900,), (987, 818), (987,))
```

Scaling and transforming our data

```
from sklearn.preprocessing import StandardScaler
import numpy as np

scalar = StandardScaler().fit(X_train[:,1:12])

def preprocessor(X):
    A = np.copy(X)
    A[:,1:12] = scalar.transform(A[:,1:12])
    return A

X_train, X_val, X_test =
preprocessor(X_train),preprocessor(X_val),preprocessor(X_test)

X_train.shape,X_val.shape,X_test.shape
```

```
((7000, 818), (987, 818), (900, 818))
```

Step 6. Applying Machine Learning Techniques to compute the Mean Square Errors and Score Metrics

Compute the mean squared error using the Linear Regression Model

Refer from this link https://encord.com/glossary/mean-square-error-mse/#:~:text=By%20squaring%20the%20differences%2C%20the,values%2C%20reflecting%20better%20overall%20performance.

```
from sklearn.metrics import mean_squared_error as mse
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

# This seems to be a good fit for the model
# NOTE: If the MSE values are large the model will likely to appear to be overfitted
lm = LinearRegression().fit(X_train,y_train)
mse(lm.predict(X_train),y_train,squared=False),
mse(lm.predict(X_val),y_val,squared=False)
```

```
(340686.8432383325, 2353038846011496.5)
```

Check the regression scores for the testing and training data.

```
lm.score(X_test,y_test)
```

```
-1.5075764864513472e+18
```

```
lm.score(X_train,y_train)
```

```
0.7489109770599006
```

In this case the training data performs significantly better than the testing data, there is overfitting for the regression model.

Compute the mean square error using KNN (K-Nearest Neighbour), Where K is any Natural Number (i.e. 1,2,..etc)

```
from sklearn.neighbors import KNeighborsRegressor

# Can set the n_neighbors for 12 for now.
# To note: As K increases, the KNN fits a smoother curve to the data. This is because a higher value of
# K reduces the edginess by taking more data into account, thus reducing the overall complexity and flexibility of the model.

#NOTE: If my MSE validation is lower than the MSE training data, this suggests there is overfitting in the training set
knn = KNeighborsRegressor(n_neighbors = 12).fit(X_train,y_train)
mse(knn.predict(X_train),y_train,squared = False), mse(knn.predict(X_val),y_val,squared=False)
```

```
(307099.1421200702, 302834.5354291967)
```

Check KNN Scores for Testing and Training Data

```
knn.score(X_test,y_test)
```

```
0.7940279909038257
```

```
knn.score(X_train,y_train)
```

```
0.7959793114644361
```

Both of these scores share similar performance, this suggests that the KNN model is best suitable model.

Compute the mean square error using the Random-Forest

```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(max_depth = 50).fit(X_train,y_train)
#Can set the max_depth to 50
# NOTE: increases the performance and makes the predictions more stable, but it also slows down the computation.
mse(rfr.predict(X_train), y_train, squared = False),mse(rfr.predict(X_val), y_val, squared = False)
```

```
(109359.19707758684, 268908.09517358575)
```

Check Random Forest Regressor Scores for Testing and Training Data

```
rfr.fit(X_train,y_train)
rfr.score(X_test,y_test)
```

```
0.8373538389986965
```

```
rfr.score(X_train,y_train)
```

```
0.9730338718106486
```

The training data performs significantly better than the testing data, there is overfitting for the Random forest model.

Compute the mean square error using the GradientBoostingRegressor

```
from sklearn.ensemble import GradientBoostingRegressor
gbr = GradientBoostingRegressor(n_estimators =
250).fit(X_train,y_train)

mse(gbr.predict(X_train), y_train, squared =
False),mse(gbr.predict(X_val), y_val, squared = False)
```

```
(215694.42126893828, 264134.6425764006)
```

Check the scores for Gradient Boosting for Testing and Training Data

```
gbr.score(X_test,y_test)
```

```
0.8414118894857592
```

```
gbr.score(X_train,y_train)
```

```
0.8993543920229342
```

The training data performs significantly better than the testing data, there is overfitting for the Random forest model.

Depends on the complexity of this model, this model will still be acceptable for good practise.

Step 7. Building Neural Networks

Installing Tensorflow package

```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\michael le\
anaconda3\lib\site-packages (2.16.1)
Requirement already satisfied: tensorflow-intel==2.16.1 in c:\users\
michael le\anaconda3\lib\site-packages (from tensorflow) (2.16.1)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in c:\users\
michael le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
c:\users\michael le\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.10.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.3.1 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (0.3.2)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.3.0)
Requirement already satisfied: packaging in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (23.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
```

```
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.31.0)
Requirement already satisfied: setuptools in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (68.2.2)
Requirement already satisfied: six>=1.12.0 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\
michael le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (4.9.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\michael
le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.62.1)
Requirement already satisfied: tensorboard<2.17,>=2.16 in c:\users\
michael le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (2.16.2)
Requirement already satisfied: keras>=3.0.0 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (3.1.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
c:\users\michael le\anaconda3\lib\site-packages (from tensorflow-
intel==2.16.1->tensorflow) (0.31.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in c:\users\
michael le\anaconda3\lib\site-packages (from tensorflow-intel==2.16.1-
>tensorflow) (1.26.4)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\michael
le\anaconda3\lib\site-packages (from astunparse>=1.6.0->tensorflow-
intel==2.16.1->tensorflow) (0.41.2)
Requirement already satisfied: rich in c:\users\michael le\anaconda3\
lib\site-packages (from keras>=3.0.0->tensorflow-intel==2.16.1-
>tensorflow) (13.3.5)
Requirement already satisfied: namex in c:\users\michael le\anaconda3\
lib\site-packages (from keras>=3.0.0->tensorflow-intel==2.16.1-
>tensorflow) (0.0.7)
Requirement already satisfied: optree in c:\users\michael le\
anaconda3\lib\site-packages (from keras>=3.0.0->tensorflow-
intel==2.16.1->tensorflow) (0.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
```

```
michael le\anaconda3\lib\site-packages (from requests<3,>=2.21.0-
>tensorflow-intel==2.16.1->tensorflow) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\michael le\
anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\michael
le\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\michael
le\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow-
intel==2.16.1->tensorflow) (2024.2.2)
Requirement already satisfied: markdown>=2.6.8 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-
intel==2.16.1->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in c:\users\michael le\anaconda3\lib\site-packages (from
tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\michael le\
anaconda3\lib\site-packages (from tensorboard<2.17,>=2.16->tensorflow-
intel==2.16.1->tensorflow) (2.2.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\michael
le\anaconda3\lib\site-packages (from werkzeug>=1.0.1-
>tensorboard<2.17,>=2.16->tensorflow-intel==2.16.1->tensorflow)
(2.1.3)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\
users\michael le\anaconda3\lib\site-packages (from rich->keras>=3.0.0-
>tensorflow-intel==2.16.1->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\
michael le\anaconda3\lib\site-packages (from rich->keras>=3.0.0-
>tensorflow-intel==2.16.1->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\michael le\
anaconda3\lib\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich-
>keras>=3.0.0->tensorflow-intel==2.16.1->tensorflow) (0.1.0)
Note: you may need to restart the kernel to use updated packages.
```

```python
import tensorflow as tf
tf.__version__
```

```
'2.16.1'
```

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import *
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.metrics import RootMeanSquaredError
from tensorflow.keras.optimizers import Adam
tf.random.set_seed(1)
np.random.seed(23)
```

Simple Neural Network

```
#Define the simple neural network
simple_nn = Sequential()

#NOTE: we input (818, ) as our input shape because it is the number of
columns we have in the final dataframe.

simple_nn.add(InputLayer((818,)))
simple_nn.add(Dense(2,'relu'))
simple_nn.add(Dense(1,'linear'))

opt = Adam(learning_rate = 0.01)
cp = ModelCheckpoint('/tmp/ckpt/checkpoint.model.keras',save_best_only
= True)
simple_nn.compile(optimizer = opt, loss = 'mse', metrics =
[RootMeanSquaredError()])
simple_nn.fit(x=X_train,y=y_train, validation_data =
(X_val,y_val),callbacks = [cp], epochs = 100)
```

```
Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 1598952570880.0000 -
root_mean_squared_error: 1264382.1250 - val_loss: 1645169344512.0000 -
val_root_mean_squared_error: 1282857.8750
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 1598531829760.0000
- root_mean_squared_error: 1264216.0000 - val_loss: 1644325371904.0000
- val_root_mean_squared_error: 1282529.0000
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 1597572120576.0000
- root_mean_squared_error: 1263836.7500 - val_loss: 1642943348736.0000
- val_root_mean_squared_error: 1281990.2500
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 750us/step - loss: 1596094939136.0000
- root_mean_squared_error: 1263252.5000 - val_loss: 1641064169472.0000
- val_root_mean_squared_error: 1281257.2500
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 1594140524544.0000
- root_mean_squared_error: 1262479.0000 - val_loss: 1638725320704.0000
- val_root_mean_squared_error: 1280344.3750
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 1591745445888.0000
- root_mean_squared_error: 1261530.2500 - val_loss: 1635961143296.0000
- val_root_mean_squared_error: 1279264.6250
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 691us/step - loss: 1588941029376.0000
- root_mean_squared_error: 1260418.3750 - val_loss: 1632800342016.0000
- val_root_mean_squared_error: 1278029.0000
```

```
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 1585754800128.0000
- root_mean_squared_error: 1259154.0000 - val_loss: 1629269000192.0000
- val_root_mean_squared_error: 1276647.0000
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 718us/step - loss: 1582211923968.0000
- root_mean_squared_error: 1257746.6250 - val_loss: 1625389924352.0000
- val_root_mean_squared_error: 1275127.2500
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 1578333372416.0000
- root_mean_squared_error: 1256204.0000 - val_loss: 1621181726720.0000
- val_root_mean_squared_error: 1273476.3750
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 1574136184832.0000
- root_mean_squared_error: 1254532.3750 - val_loss: 1616661577728.0000
- val_root_mean_squared_error: 1271700.8750
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 661us/step - loss: 1569638318080.0000
- root_mean_squared_error: 1252738.6250 - val_loss: 1611844419584.0000
- val_root_mean_squared_error: 1269805.8750
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 668us/step - loss: 1564852486144.0000
- root_mean_squared_error: 1250827.1250 - val_loss: 1606742310912.0000
- val_root_mean_squared_error: 1267795.7500
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 675us/step - loss: 1559790747648.0000
- root_mean_squared_error: 1248802.2500 - val_loss: 1601367441408.0000
- val_root_mean_squared_error: 1265674.6250
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1554464374784.0000
- root_mean_squared_error: 1246667.8750 - val_loss: 1595729641472.0000
- val_root_mean_squared_error: 1263445.8750
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 658us/step - loss: 1548882935808.0000
- root_mean_squared_error: 1244427.3750 - val_loss: 1589837430784.0000
- val_root_mean_squared_error: 1261112.5000
Epoch 17/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 670us/step - loss: 1543054557184.0000
- root_mean_squared_error: 1242083.5000 - val_loss: 1583698935808.0000
- val_root_mean_squared_error: 1258677.0000
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 1536986447872.0000
- root_mean_squared_error: 1239638.5000 - val_loss: 1577321627648.0000
- val_root_mean_squared_error: 1256141.6250
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 1530685423616.0000
- root_mean_squared_error: 1237094.5000 - val_loss: 1570710880256.0000
- val_root_mean_squared_error: 1253508.0000
Epoch 20/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1524158038016.0000
- root_mean_squared_error: 1234453.5000 - val_loss: 1563873247232.0000
- val_root_mean_squared_error: 1250778.2500
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 658us/step - loss: 1517409140736.0000
- root_mean_squared_error: 1231717.0000 - val_loss: 1556813316096.0000
- val_root_mean_squared_error: 1247953.3750
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 654us/step - loss: 1510443843584.0000
- root_mean_squared_error: 1228886.3750 - val_loss: 1549536854016.0000
- val_root_mean_squared_error: 1245035.2500
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 674us/step - loss: 1503267389440.0000
- root_mean_squared_error: 1225963.0000 - val_loss: 1542047399936.0000
- val_root_mean_squared_error: 1242024.5000
Epoch 24/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 1495883710464.0000 -
root_mean_squared_error: 1222948.0000 - val_loss: 1534350721024.0000 -
val_root_mean_squared_error: 1238922.7500
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 1488297525248.0000
- root_mean_squared_error: 1219842.3750 - val_loss: 1526449307648.0000
- val_root_mean_squared_error: 1235730.5000
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 1480511979520.0000
- root_mean_squared_error: 1216647.1250 - val_loss: 1518348009472.0000
- val_root_mean_squared_error: 1232448.8750
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 1472531267584.0000
- root_mean_squared_error: 1213362.8750 - val_loss: 1510050365440.0000
- val_root_mean_squared_error: 1229078.6250
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 687us/step - loss: 1464359190528.0000
- root_mean_squared_error: 1209990.6250 - val_loss: 1501559783424.0000
- val_root_mean_squared_error: 1225620.3750
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 691us/step - loss: 1455998500864.0000
- root_mean_squared_error: 1206530.8750 - val_loss: 1492880064512.0000
- val_root_mean_squared_error: 1222074.8750
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 1447453392896.0000
- root_mean_squared_error: 1202984.5000 - val_loss: 1484014747648.0000
- val_root_mean_squared_error: 1218443.0000
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 688us/step - loss: 1438727274496.0000
- root_mean_squared_error: 1199352.1250 - val_loss: 1474966716416.0000
- val_root_mean_squared_error: 1214725.1250
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 1429822636032.0000
```

```
- root_mean_squared_error: 1195634.0000 - val_loss: 1465740296192.0000
- val_root_mean_squared_error: 1210922.0000
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 1420743540736.0000
- root_mean_squared_error: 1191831.2500 - val_loss: 1456337846272.0000
- val_root_mean_squared_error: 1207034.1250
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 667us/step - loss: 1411493003264.0000
- root_mean_squared_error: 1187944.1250 - val_loss: 1446763429888.0000
- val_root_mean_squared_error: 1203062.2500
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 668us/step - loss: 1402073776128.0000
- root_mean_squared_error: 1183972.8750 - val_loss: 1437019799552.0000
- val_root_mean_squared_error: 1199006.6250
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 1392490184704.0000
- root_mean_squared_error: 1179918.7500 - val_loss: 1427111149568.0000
- val_root_mean_squared_error: 1194868.1250
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 1382744719360.0000
- root_mean_squared_error: 1175781.6250 - val_loss: 1417039708160.0000
- val_root_mean_squared_error: 1190647.0000
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 1372840919040.0000
- root_mean_squared_error: 1171562.3750 - val_loss: 1406809538560.0000
- val_root_mean_squared_error: 1186343.7500
Epoch 39/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1362781536256.0000
- root_mean_squared_error: 1167261.3750 - val_loss: 1396423917568.0000
- val_root_mean_squared_error: 1181959.2500
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1352570503168.0000
- root_mean_squared_error: 1162879.1250 - val_loss: 1385886384128.0000
- val_root_mean_squared_error: 1177493.8750
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 705us/step - loss: 1342211096576.0000
- root_mean_squared_error: 1158416.2500 - val_loss: 1375200083968.0000
- val_root_mean_squared_error: 1172948.1250
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 674us/step - loss: 1331706462208.0000
- root_mean_squared_error: 1153873.1250 - val_loss: 1364368687104.0000
- val_root_mean_squared_error: 1168322.5000
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 1321059614720.0000
- root_mean_squared_error: 1149250.3750 - val_loss: 1353395339264.0000
- val_root_mean_squared_error: 1163617.3750
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 1310274224128.0000
- root_mean_squared_error: 1144548.2500 - val_loss: 1342283710464.0000
```

```
- val_root_mean_squared_error: 1158833.7500
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 1299353960448.0000
- root_mean_squared_error: 1139767.6250 - val_loss: 1331037863936.0000
- val_root_mean_squared_error: 1153972.0000
Epoch 46/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 671us/step - loss: 1288302362624.0000
- root_mean_squared_error: 1134908.8750 - val_loss: 1319660814336.0000
- val_root_mean_squared_error: 1149032.3750
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 1277122576384.0000
- root_mean_squared_error: 1129972.6250 - val_loss: 1308156362752.0000
- val_root_mean_squared_error: 1144016.1250
Epoch 48/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 1265817878528.0000
- root_mean_squared_error: 1124959.1250 - val_loss: 1296527654912.0000
- val_root_mean_squared_error: 1138923.0000
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 1254392463360.0000
- root_mean_squared_error: 1119869.3750 - val_loss: 1284778754048.0000
- val_root_mean_squared_error: 1133754.0000
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1242849476608.0000
- root_mean_squared_error: 1114703.5000 - val_loss: 1272913854464.0000
- val_root_mean_squared_error: 1128509.8750
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 675us/step - loss: 1231192195072.0000
- root_mean_squared_error: 1109462.1250 - val_loss: 1260935839744.0000
- val_root_mean_squared_error: 1123191.0000
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 667us/step - loss: 1219425206272.0000
- root_mean_squared_error: 1104146.1250 - val_loss: 1248848904192.0000
- val_root_mean_squared_error: 1117798.0000
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 1207551393792.0000
- root_mean_squared_error: 1098755.8750 - val_loss: 1236657111040.0000
- val_root_mean_squared_error: 1112331.6250
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 1195574820864.0000
- root_mean_squared_error: 1093292.0000 - val_loss: 1224363474944.0000
- val_root_mean_squared_error: 1106792.3750
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 1183498764288.0000
- root_mean_squared_error: 1087755.0000 - val_loss: 1211972714496.0000
- val_root_mean_squared_error: 1101181.0000
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 1171327811584.0000
- root_mean_squared_error: 1082145.7500 - val_loss: 1199488237568.0000
- val_root_mean_squared_error: 1095498.2500
```

```
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 1159065501696.0000
- root_mean_squared_error: 1076464.8750 - val_loss: 1186914369536.0000
- val_root_mean_squared_error: 1089744.7500
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 1146715111424.0000
- root_mean_squared_error: 1070712.6250 - val_loss: 1174254780416.0000
- val_root_mean_squared_error: 1083921.0000
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 1134280835072.0000
- root_mean_squared_error: 1064890.0000 - val_loss: 1161513402368.0000
- val_root_mean_squared_error: 1078028.0000
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 1121766735872.0000
- root_mean_squared_error: 1058997.6250 - val_loss: 1148694691840.0000
- val_root_mean_squared_error: 1072066.5000
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 1109177008128.0000
- root_mean_squared_error: 1053036.2500 - val_loss: 1135802580992.0000
- val_root_mean_squared_error: 1066037.1250
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 1096515321856.0000
- root_mean_squared_error: 1047006.6875 - val_loss: 1122841264128.0000
- val_root_mean_squared_error: 1059940.7500
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - loss: 1083785347072.0000
- root_mean_squared_error: 1040909.3125 - val_loss: 1109814542336.0000
- val_root_mean_squared_error: 1053778.1250
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 693us/step - loss: 1070991802368.0000
- root_mean_squared_error: 1034745.3125 - val_loss: 1096727003136.0000
- val_root_mean_squared_error: 1047550.1250
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 689us/step - loss: 1058138161152.0000
- root_mean_squared_error: 1028515.1250 - val_loss: 1083582578688.0000
- val_root_mean_squared_error: 1041257.6250
Epoch 66/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 702us/step - loss: 1045229076480.0000
- root_mean_squared_error: 1022219.8125 - val_loss: 1070385922048.0000
- val_root_mean_squared_error: 1034901.4375
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 1032268611584.0000
- root_mean_squared_error: 1015860.1875 - val_loss: 1057140899840.0000
- val_root_mean_squared_error: 1028482.5000
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - loss: 1019260764160.0000
- root_mean_squared_error: 1009437.0625 - val_loss: 1043852623872.0000
- val_root_mean_squared_error: 1022002.0000
Epoch 69/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 712us/step - loss: 1006210449408.0000
- root_mean_squared_error: 1002951.5625 - val_loss: 1030524174336.0000
- val_root_mean_squared_error: 1015460.3750
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 993120288768.0000 -
root_mean_squared_error: 996403.8750 - val_loss: 1017160925184.0000 -
val_root_mean_squared_error: 1008859.0000
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 979995918336.0000 -
root_mean_squared_error: 989795.5000 - val_loss: 1003766874112.0000 -
val_root_mean_squared_error: 1002198.7500
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 719us/step - loss: 966840942592.0000 -
root_mean_squared_error: 983127.1875 - val_loss: 990346674176.0000 -
val_root_mean_squared_error: 995480.7500
Epoch 73/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 953660407808.0000 -
root_mean_squared_error: 976400.1875 - val_loss: 976904454144.0000 -
val_root_mean_squared_error: 988705.8750
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 940458115072.0000 -
root_mean_squared_error: 969615.3125 - val_loss: 963445129216.0000 -
val_root_mean_squared_error: 981875.5000
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 691us/step - loss: 927238651904.0000 -
root_mean_squared_error: 962773.6250 - val_loss: 949973352448.0000 -
val_root_mean_squared_error: 974990.8750
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 689us/step - loss: 914006474752.0000 -
root_mean_squared_error: 955876.3125 - val_loss: 936493842432.0000 -
val_root_mean_squared_error: 968053.1875
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 900766564352.0000 -
root_mean_squared_error: 948924.7500 - val_loss: 923010334720.0000 -
val_root_mean_squared_error: 961063.2500
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 714us/step - loss: 887522197504.0000 -
root_mean_squared_error: 941919.5625 - val_loss: 909527744512.0000 -
val_root_mean_squared_error: 954022.5625
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 874278748160.0000 -
root_mean_squared_error: 934862.2500 - val_loss: 896050724864.0000 -
val_root_mean_squared_error: 946932.4375
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 703us/step - loss: 861040345088.0000 -
root_mean_squared_error: 927754.0000 - val_loss: 882584387584.0000 -
val_root_mean_squared_error: 939794.3750
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 708us/step - loss: 847811837952.0000 -
```

```
root_mean_squared_error: 920596.2500 - val_loss: 869133451264.0000 -
val_root_mean_squared_error: 932609.8750
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 834597879808.0000 -
root_mean_squared_error: 913390.2500 - val_loss: 855702110208.0000 -
val_root_mean_squared_error: 925380.0625
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 821402664960.0000 -
root_mean_squared_error: 906137.3125 - val_loss: 842295738368.0000 -
val_root_mean_squared_error: 918106.9375
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 808231698432.0000 -
root_mean_squared_error: 898839.2500 - val_loss: 828918661120.0000 -
val_root_mean_squared_error: 910791.6875
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 795088781312.0000 -
root_mean_squared_error: 891497.1875 - val_loss: 815576055808.0000 -
val_root_mean_squared_error: 903436.1875
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 708us/step - loss: 781979156480.0000 -
root_mean_squared_error: 884112.9375 - val_loss: 802272509952.0000 -
val_root_mean_squared_error: 896042.0625
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 768906952704.0000 -
root_mean_squared_error: 876687.8125 - val_loss: 789012676608.0000 -
val_root_mean_squared_error: 888610.8750
Epoch 88/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 755877085184.0000 -
root_mean_squared_error: 869223.5625 - val_loss: 775801405440.0000 -
val_root_mean_squared_error: 881144.5000
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 701us/step - loss: 742894403584.0000 -
root_mean_squared_error: 861722.0000 - val_loss: 762643546112.0000 -
val_root_mean_squared_error: 873644.7500
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 707us/step - loss: 729963298816.0000 -
root_mean_squared_error: 854184.6250 - val_loss: 749543882752.0000 -
val_root_mean_squared_error: 866113.7500
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 717088423936.0000 -
root_mean_squared_error: 846613.4375 - val_loss: 736507592704.0000 -
val_root_mean_squared_error: 858553.2500
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 704275152896.0000 -
root_mean_squared_error: 839010.5000 - val_loss: 723539853312.0000 -
val_root_mean_squared_error: 850965.8750
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 691528400896.0000 -
root_mean_squared_error: 831378.0000 - val_loss: 710645317632.0000 -
```

```
val_root_mean_squared_error: 843353.6250
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 722us/step - loss: 678852952064.0000 -
root_mean_squared_error: 823718.0625 - val_loss: 697829294080.0000 -
val_root_mean_squared_error: 835718.7500
Epoch 95/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 666252935168.0000 -
root_mean_squared_error: 816032.3750 - val_loss: 685095911424.0000 -
val_root_mean_squared_error: 828063.3750
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 653733658624.0000 -
root_mean_squared_error: 808323.5000 - val_loss: 672451067904.0000 -
val_root_mean_squared_error: 820390.3125
Epoch 97/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 641300103168.0000 -
root_mean_squared_error: 800593.8750 - val_loss: 659898630144.0000 -
val_root_mean_squared_error: 812701.6875
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 628956528640.0000 -
root_mean_squared_error: 792845.5625 - val_loss: 647444234240.0000 -
val_root_mean_squared_error: 805000.3125
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 724us/step - loss: 616708112384.0000 -
root_mean_squared_error: 785081.3750 - val_loss: 635092533248.0000 -
val_root_mean_squared_error: 797288.8750
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 604559507456.0000 -
root_mean_squared_error: 777303.7500 - val_loss: 622848770048.0000 -
val_root_mean_squared_error: 789570.3125

<keras.src.callbacks.history.History at 0x2355ce913d0>

from tensorflow.keras.models import load_model
simple_nn = load_model('/tmp/ckpt/checkpoint.model.keras')

#Compute the mean square errors from the simple neural network
#We want to ensure our simple neural network improves overfitting for
both the training and validation data.
#Test the learning rate to see different results (relatively a small
learning value will be good enough).
mse(simple_nn.predict(X_train), y_train, squared =
False),mse(simple_nn.predict(X_val), y_val, squared = False)

219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 507us/step
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 465us/step

(791370.3336280907, 789570.3421134006)

history = simple_nn.fit(x=X_train,y=y_train, validation_data =
(X_val,y_val),callbacks = [cp], batch_size = 32, epochs = 100, verbose
```

```
= 1)
# Get training and test loss histories
training_loss = history.history['loss']
test_loss = history.history['val_loss']

# Create count of the number of epochs
epoch_count = range(1, len(training_loss) + 1)
```

```
Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 592515891200.0000 -
root_mean_squared_error: 769515.6250 - val_loss: 610717532160.0000 -
val_root_mean_squared_error: 781847.4375
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 580581457920.0000 -
root_mean_squared_error: 761719.5625 - val_loss: 598704259072.0000 -
val_root_mean_squared_error: 774123.5625
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 673us/step - loss: 568761450496.0000 -
root_mean_squared_error: 753918.6875 - val_loss: 586813341696.0000 -
val_root_mean_squared_error: 766401.5625
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 667us/step - loss: 557061111808.0000 -
root_mean_squared_error: 746116.3750 - val_loss: 575050022912.0000 -
val_root_mean_squared_error: 758684.9375
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 545484308480.0000 -
root_mean_squared_error: 738315.1875 - val_loss: 563418955776.0000 -
val_root_mean_squared_error: 750976.8750
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 667us/step - loss: 534036217856.0000 -
root_mean_squared_error: 730518.6875 - val_loss: 551924858880.0000 -
val_root_mean_squared_error: 743280.9375
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 522721394688.0000 -
root_mean_squared_error: 722730.3125 - val_loss: 540572483584.0000 -
val_root_mean_squared_error: 735600.7500
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 511543967744.0000 -
root_mean_squared_error: 714953.0000 - val_loss: 529366482944.0000 -
val_root_mean_squared_error: 727939.8125
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 698us/step - loss: 500509900800.0000 -
root_mean_squared_error: 707191.5625 - val_loss: 518311870464.0000 -
val_root_mean_squared_error: 720302.3750
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 489622765568.0000 -
root_mean_squared_error: 699448.8750 - val_loss: 507413561344.0000 -
val_root_mean_squared_error: 712692.6875
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 478887444480.0000 -
```

```
root_mean_squared_error: 691729.2500 - val_loss: 496675192832.0000 -
val_root_mean_squared_error: 705114.0625
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 468308099072.0000 -
root_mean_squared_error: 684036.3750 - val_loss: 486101843968.0000 -
val_root_mean_squared_error: 697571.2500
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 674us/step - loss: 457889546240.0000 -
root_mean_squared_error: 676374.8750 - val_loss: 475697741824.0000 -
val_root_mean_squared_error: 690068.5000
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 715us/step - loss: 447635718144.0000 -
root_mean_squared_error: 668748.6875 - val_loss: 465467375616.0000 -
val_root_mean_squared_error: 682610.5000
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 437551136768.0000 -
root_mean_squared_error: 661162.4375 - val_loss: 455415070720.0000 -
val_root_mean_squared_error: 675201.6875
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 427639865344.0000 -
root_mean_squared_error: 653620.6875 - val_loss: 445544726528.0000 -
val_root_mean_squared_error: 667846.8125
Epoch 17/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 670us/step - loss: 417906032640.0000 -
root_mean_squared_error: 646128.0625 - val_loss: 435860701184.0000 -
val_root_mean_squared_error: 660550.9375
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 408353865728.0000 -
root_mean_squared_error: 638689.8125 - val_loss: 426366337024.0000 -
val_root_mean_squared_error: 653318.6875
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 687us/step - loss: 398986477568.0000 -
root_mean_squared_error: 631310.1875 - val_loss: 417065926656.0000 -
val_root_mean_squared_error: 646155.3125
Epoch 20/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 389808193536.0000 -
root_mean_squared_error: 623994.7500 - val_loss: 407962877952.0000 -
val_root_mean_squared_error: 639066.0000
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 380822355968.0000 -
root_mean_squared_error: 616748.5625 - val_loss: 399060795392.0000 -
val_root_mean_squared_error: 632056.0000
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 672us/step - loss: 372032765952.0000 -
root_mean_squared_error: 609577.1250 - val_loss: 390363217920.0000 -
val_root_mean_squared_error: 625130.8750
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 363442405376.0000 -
root_mean_squared_error: 602485.6875 - val_loss: 381872504832.0000 -
```

```
val_root_mean_squared_error: 618295.2500
Epoch 24/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 355054059520.0000 -
root_mean_squared_error: 595479.4375 - val_loss: 373591638016.0000 -
val_root_mean_squared_error: 611554.6875
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 346870710272.0000 -
root_mean_squared_error: 588563.8750 - val_loss: 365523501056.0000 -
val_root_mean_squared_error: 604914.7500
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 338894782464.0000 -
root_mean_squared_error: 581744.4375 - val_loss: 357670191104.0000 -
val_root_mean_squared_error: 598380.5625
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 678us/step - loss: 331129159680.0000 -
root_mean_squared_error: 575027.0000 - val_loss: 350034853888.0000 -
val_root_mean_squared_error: 591958.1875
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 676us/step - loss: 323576070144.0000 -
root_mean_squared_error: 568417.1875 - val_loss: 342618144768.0000 -
val_root_mean_squared_error: 585651.9375
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 316236759040.0000 -
root_mean_squared_error: 561919.9375 - val_loss: 335423143936.0000 -
val_root_mean_squared_error: 579468.3125
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 309114175488.0000 -
root_mean_squared_error: 555541.8125 - val_loss: 328451129344.0000 -
val_root_mean_squared_error: 573412.3750
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 302209499136.0000 -
root_mean_squared_error: 549288.0625 - val_loss: 321702658048.0000 -
val_root_mean_squared_error: 567488.6875
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 678us/step - loss: 295523418112.0000 -
root_mean_squared_error: 543163.6250 - val_loss: 315178483712.0000 -
val_root_mean_squared_error: 561702.0625
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 289056915456.0000 -
root_mean_squared_error: 537173.9375 - val_loss: 308879851520.0000 -
val_root_mean_squared_error: 556058.0000
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 679us/step - loss: 282810744832.0000 -
root_mean_squared_error: 531324.1875 - val_loss: 302806106112.0000 -
val_root_mean_squared_error: 550560.3125
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 689us/step - loss: 276784775168.0000 -
root_mean_squared_error: 525619.0000 - val_loss: 296957968384.0000 -
val_root_mean_squared_error: 545214.0000
```

```
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 270979727360.0000 -
root_mean_squared_error: 520063.8750 - val_loss: 291334586368.0000 -
val_root_mean_squared_error: 540022.8750
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 265394700288.0000 -
root_mean_squared_error: 514662.7188 - val_loss: 285935140864.0000 -
val_root_mean_squared_error: 534990.6875
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 260029186048.0000 -
root_mean_squared_error: 509420.0312 - val_loss: 280756027392.0000 -
val_root_mean_squared_error: 530118.5625
Epoch 39/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 254881890304.0000 -
root_mean_squared_error: 504339.4375 - val_loss: 275797245952.0000 -
val_root_mean_squared_error: 525410.8750
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 669us/step - loss: 249950830592.0000 -
root_mean_squared_error: 499423.9688 - val_loss: 271057502208.0000 -
val_root_mean_squared_error: 520871.0625
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 245234466816.0000 -
root_mean_squared_error: 494676.9688 - val_loss: 266533715968.0000 -
val_root_mean_squared_error: 516500.4062
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 687us/step - loss: 240730079232.0000 -
root_mean_squared_error: 490100.5938 - val_loss: 262223183872.0000 -
val_root_mean_squared_error: 512300.7188
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 236435013632.0000 -
root_mean_squared_error: 485696.9062 - val_loss: 258122579968.0000 -
val_root_mean_squared_error: 508272.9375
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 232346189824.0000 -
root_mean_squared_error: 481467.5312 - val_loss: 254227939328.0000 -
val_root_mean_squared_error: 504417.2812
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 228459872256.0000 -
root_mean_squared_error: 477413.1562 - val_loss: 250535116800.0000 -
val_root_mean_squared_error: 500733.5938
Epoch 46/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 224772030464.0000 -
root_mean_squared_error: 473534.0000 - val_loss: 247039574016.0000 -
val_root_mean_squared_error: 497221.1875
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 678us/step - loss: 221278142464.0000 -
root_mean_squared_error: 469829.5938 - val_loss: 243735838720.0000 -
val_root_mean_squared_error: 493878.1562
Epoch 48/100
```

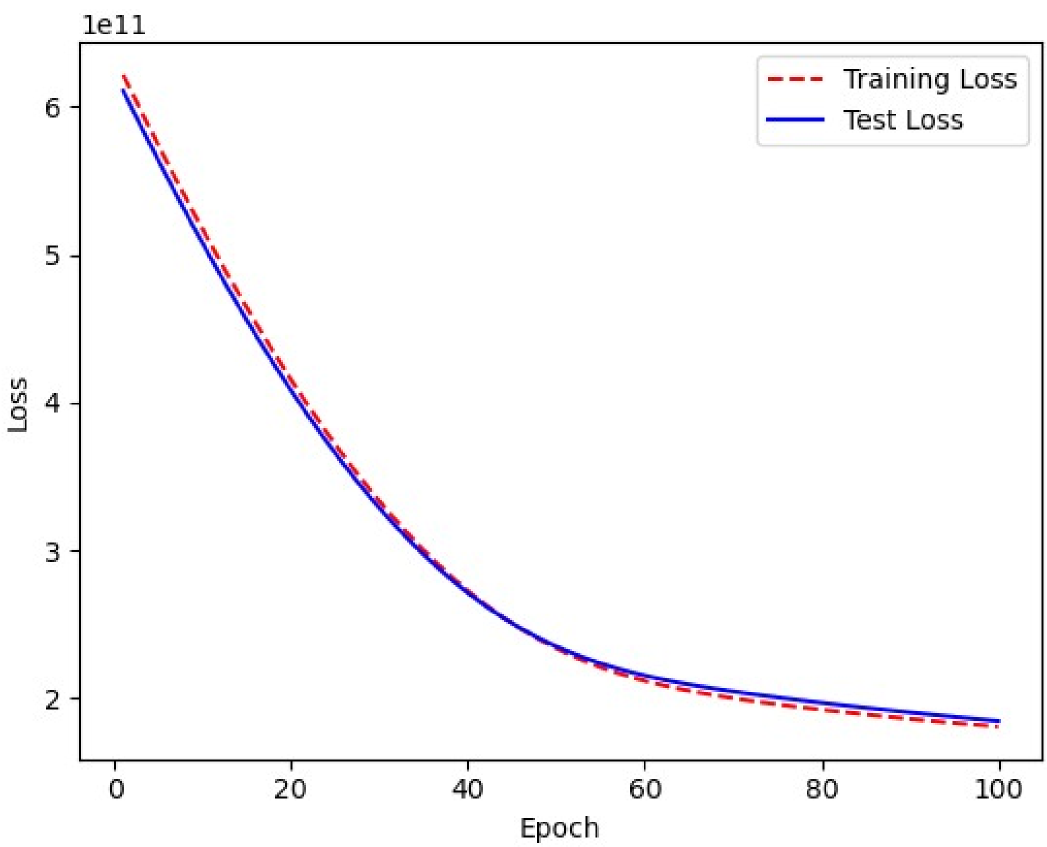
```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 217973506048.0000 -
root_mean_squared_error: 466299.1250 - val_loss: 240617881600.0000 -
val_root_mean_squared_error: 490701.8750
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 214852550656.0000 -
root_mean_squared_error: 462940.5000 - val_loss: 237678346240.0000 -
val_root_mean_squared_error: 487688.0312
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 693us/step - loss: 211909672960.0000 -
root_mean_squared_error: 459751.3750 - val_loss: 234912808960.0000 -
val_root_mean_squared_error: 484835.1250
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 209138974720.0000 -
root_mean_squared_error: 456728.8125 - val_loss: 232314093568.0000 -
val_root_mean_squared_error: 482138.6250
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 684us/step - loss: 206533902336.0000 -
root_mean_squared_error: 453868.9688 - val_loss: 229873909760.0000 -
val_root_mean_squared_error: 479592.5000
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 204088016896.0000 -
root_mean_squared_error: 451167.7500 - val_loss: 227586375680.0000 -
val_root_mean_squared_error: 477193.0312
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 682us/step - loss: 201794174976.0000 -
root_mean_squared_error: 448620.0000 - val_loss: 225444200448.0000 -
val_root_mean_squared_error: 474934.8125
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 679us/step - loss: 199645134848.0000 -
root_mean_squared_error: 446220.2500 - val_loss: 223439388672.0000 -
val_root_mean_squared_error: 472811.3750
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 197633392640.0000 -
root_mean_squared_error: 443962.4062 - val_loss: 221563961344.0000 -
val_root_mean_squared_error: 470816.1562
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 195751346176.0000 -
root_mean_squared_error: 441840.0312 - val_loss: 219809775616.0000 -
val_root_mean_squared_error: 468942.0625
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 193991131136.0000 -
root_mean_squared_error: 439846.0625 - val_loss: 218168459264.0000 -
val_root_mean_squared_error: 467181.6562
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 192344686592.0000 -
root_mean_squared_error: 437973.0938 - val_loss: 216631771136.0000 -
val_root_mean_squared_error: 465527.3125
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 679us/step - loss: 190804115456.0000 -
```

```
root_mean_squared_error: 436213.5312 - val_loss: 215191306240.0000 -
val_root_mean_squared_error: 463971.1875
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 688us/step - loss: 189361520640.0000 -
root_mean_squared_error: 434559.6562 - val_loss: 213839036416.0000 -
val_root_mean_squared_error: 462505.5625
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 688us/step - loss: 188008890368.0000 -
root_mean_squared_error: 433003.3750 - val_loss: 212566818816.0000 -
val_root_mean_squared_error: 461122.5312
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 186738556928.0000 -
root_mean_squared_error: 431536.9062 - val_loss: 211366363136.0000 -
val_root_mean_squared_error: 459813.7500
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 691us/step - loss: 185542852608.0000 -
root_mean_squared_error: 430152.1250 - val_loss: 210230951936.0000 -
val_root_mean_squared_error: 458572.5625
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 687us/step - loss: 184414814208.0000 -
root_mean_squared_error: 428841.7500 - val_loss: 209153540096.0000 -
val_root_mean_squared_error: 457391.8438
Epoch 66/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 183347707904.0000 -
root_mean_squared_error: 427598.5312 - val_loss: 208127688704.0000 -
val_root_mean_squared_error: 456264.9688
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 182335094784.0000 -
root_mean_squared_error: 426415.5000 - val_loss: 207146762240.0000 -
val_root_mean_squared_error: 455185.0000
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 677us/step - loss: 181370798080.0000 -
root_mean_squared_error: 425285.9062 - val_loss: 206206074880.0000 -
val_root_mean_squared_error: 454147.1875
Epoch 69/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 180449984512.0000 -
root_mean_squared_error: 424204.4062 - val_loss: 205300793344.0000 -
val_root_mean_squared_error: 453146.4375
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 689us/step - loss: 179567673344.0000 -
root_mean_squared_error: 423165.5000 - val_loss: 204426428416.0000 -
val_root_mean_squared_error: 452178.0000
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 178719490048.0000 -
root_mean_squared_error: 422164.3438 - val_loss: 203578916864.0000 -
val_root_mean_squared_error: 451237.5312
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 177901305856.0000 -
root_mean_squared_error: 421196.2812 - val_loss: 202755473408.0000 -
```

```
val_root_mean_squared_error: 450322.1250
Epoch 73/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 177110204416.0000 -
root_mean_squared_error: 420258.0312 - val_loss: 201953312768.0000 -
val_root_mean_squared_error: 449428.8750
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 176343285760.0000 -
root_mean_squared_error: 419346.4062 - val_loss: 201170190336.0000 -
val_root_mean_squared_error: 448555.2812
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 175598141440.0000 -
root_mean_squared_error: 418458.6875 - val_loss: 200404058112.0000 -
val_root_mean_squared_error: 447699.2188
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 174872641536.0000 -
root_mean_squared_error: 417592.4375 - val_loss: 199653310464.0000 -
val_root_mean_squared_error: 446858.9688
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 715us/step - loss: 174164918272.0000 -
root_mean_squared_error: 416745.6250 - val_loss: 198916685824.0000 -
val_root_mean_squared_error: 446033.1562
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 681us/step - loss: 173472972800.0000 -
root_mean_squared_error: 415915.8750 - val_loss: 198192480256.0000 -
val_root_mean_squared_error: 445219.9375
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 172795363328.0000 -
root_mean_squared_error: 415101.6875 - val_loss: 197480398848.0000 -
val_root_mean_squared_error: 444419.0312
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 172131811328.0000 -
root_mean_squared_error: 414302.7188 - val_loss: 196779737088.0000 -
val_root_mean_squared_error: 443629.7500
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 706us/step - loss: 171480776704.0000 -
root_mean_squared_error: 413517.2188 - val_loss: 196089446400.0000 -
val_root_mean_squared_error: 442850.8750
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 723us/step - loss: 170841735168.0000 -
root_mean_squared_error: 412744.6562 - val_loss: 195407855616.0000 -
val_root_mean_squared_error: 442080.5938
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 170214424576.0000 -
root_mean_squared_error: 411984.7812 - val_loss: 194732802048.0000 -
val_root_mean_squared_error: 441316.4688
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 169598238720.0000 -
root_mean_squared_error: 411236.9688 - val_loss: 194067300352.0000 -
val_root_mean_squared_error: 440561.9375
```

```
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 168992718848.0000 -
root_mean_squared_error: 410500.7188 - val_loss: 193411203072.0000 -
val_root_mean_squared_error: 439816.9062
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 168397537280.0000 -
root_mean_squared_error: 409775.6562 - val_loss: 192762707968.0000 -
val_root_mean_squared_error: 439079.3125
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 167812284416.0000 -
root_mean_squared_error: 409061.4062 - val_loss: 192121880576.0000 -
val_root_mean_squared_error: 438349.2812
Epoch 88/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 705us/step - loss: 167236632576.0000 -
root_mean_squared_error: 408357.5938 - val_loss: 191488851968.0000 -
val_root_mean_squared_error: 437627.0000
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 683us/step - loss: 166670270464.0000 -
root_mean_squared_error: 407663.9062 - val_loss: 190858969088.0000 -
val_root_mean_squared_error: 436907.1562
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 166113083392.0000 -
root_mean_squared_error: 406980.2500 - val_loss: 190237687808.0000 -
val_root_mean_squared_error: 436196.0000
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 718us/step - loss: 165564334080.0000 -
root_mean_squared_error: 406305.8438 - val_loss: 189623992320.0000 -
val_root_mean_squared_error: 435492.4688
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 750us/step - loss: 165023694848.0000 -
root_mean_squared_error: 405640.2500 - val_loss: 189017784320.0000 -
val_root_mean_squared_error: 434796.4375
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 164491214848.0000 -
root_mean_squared_error: 404983.5938 - val_loss: 188420096000.0000 -
val_root_mean_squared_error: 434109.1562
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 694us/step - loss: 163967156224.0000 -
root_mean_squared_error: 404336.2188 - val_loss: 187830779904.0000 -
val_root_mean_squared_error: 433430.4375
Epoch 95/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 163451305984.0000 -
root_mean_squared_error: 403698.0000 - val_loss: 187249819648.0000 -
val_root_mean_squared_error: 432760.3125
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 694us/step - loss: 162943549440.0000 -
root_mean_squared_error: 403068.7500 - val_loss: 186676969472.0000 -
val_root_mean_squared_error: 432098.5938
Epoch 97/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 162443640832.0000 -
root_mean_squared_error: 402448.2500 - val_loss: 186111541248.0000 -
val_root_mean_squared_error: 431444.4375
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 705us/step - loss: 161951318016.0000 -
root_mean_squared_error: 401836.2188 - val_loss: 185553895424.0000 -
val_root_mean_squared_error: 430798.3750
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 698us/step - loss: 161466384384.0000 -
root_mean_squared_error: 401232.4375 - val_loss: 185004212224.0000 -
val_root_mean_squared_error: 430160.5938
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 160989003776.0000 -
root_mean_squared_error: 400637.1562 - val_loss: 184462393344.0000 -
val_root_mean_squared_error: 429531.0000

# Visualize loss history for the simple neural network
plt.plot(epoch_count, training_loss, 'r--')
plt.plot(epoch_count, test_loss, 'b-')
plt.legend(['Training Loss', 'Test Loss'])
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show();
```



Medium neural network

```
medium_nn = Sequential()
medium_nn.add(InputLayer((818,)))
medium_nn.add(Dense(32, 'relu'))
medium_nn.add(Dense(16, 'relu'))
medium_nn.add(Dense(1, 'linear'))

opt = Adam(learning_rate=.1)
cp = ModelCheckpoint('/tmp/ckpt/checkpoint.model.keras',
save_best_only=True)
medium_nn.compile(optimizer=opt, loss='mse',
metrics=[RootMeanSquaredError()])
medium_nn.fit(x=X_train, y=y_train, validation_data=(X_val, y_val),
callbacks=[cp], epochs=100)
```

```
Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 903190544384.0000 -
root_mean_squared_error: 929812.6875 - val_loss: 141236944896.0000 -
val_root_mean_squared_error: 375997.1875
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 124218621952.0000 -
root_mean_squared_error: 351857.3750 - val_loss: 125398876160.0000 -
val_root_mean_squared_error: 354357.0938
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 754us/step - loss: 113681047552.0000 -
root_mean_squared_error: 336460.4375 - val_loss: 116372447232.0000 -
val_root_mean_squared_error: 341417.8750
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 783us/step - loss: 106807263232.0000 -
root_mean_squared_error: 326097.3438 - val_loss: 108625805312.0000 -
val_root_mean_squared_error: 329917.6250
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 100670816256.0000 -
root_mean_squared_error: 316555.2812 - val_loss: 102182690816.0000 -
val_root_mean_squared_error: 320036.4375
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 758us/step - loss: 95836569600.0000 -
root_mean_squared_error: 308808.3125 - val_loss: 97313497088.0000 -
val_root_mean_squared_error: 312354.9688
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 757us/step - loss: 92177391616.0000 -
root_mean_squared_error: 302811.0625 - val_loss: 93891166208.0000 -
val_root_mean_squared_error: 306834.8125
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 89611296768.0000 -
root_mean_squared_error: 298544.3125 - val_loss: 91584528384.0000 -
val_root_mean_squared_error: 303059.2812
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 755us/step - loss: 87848361984.0000 -
```

```
root_mean_squared_error: 295587.6875 - val_loss: 90043088896.0000 -
val_root_mean_squared_error: 300512.3750
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 86565273600.0000 -
root_mean_squared_error: 293431.1875 - val_loss: 89076596736.0000 -
val_root_mean_squared_error: 298907.4062
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 85517811712.0000 -
root_mean_squared_error: 291655.5312 - val_loss: 88231976960.0000 -
val_root_mean_squared_error: 297497.6875
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 768us/step - loss: 84589797376.0000 -
root_mean_squared_error: 290077.6250 - val_loss: 87614808064.0000 -
val_root_mean_squared_error: 296470.1562
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 763us/step - loss: 83747667968.0000 -
root_mean_squared_error: 288632.5625 - val_loss: 87136845824.0000 -
val_root_mean_squared_error: 295669.0625
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 763us/step - loss: 83002081280.0000 -
root_mean_squared_error: 287350.8750 - val_loss: 86726385664.0000 -
val_root_mean_squared_error: 294978.0938
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 760us/step - loss: 82243584000.0000 -
root_mean_squared_error: 286038.4375 - val_loss: 86399778816.0000 -
val_root_mean_squared_error: 294422.6875
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 776us/step - loss: 81590452224.0000 -
root_mean_squared_error: 284909.0312 - val_loss: 86211895296.0000 -
val_root_mean_squared_error: 294102.0000
Epoch 17/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 754us/step - loss: 80929701888.0000 -
root_mean_squared_error: 283761.5312 - val_loss: 86011641856.0000 -
val_root_mean_squared_error: 293757.7812
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 780us/step - loss: 80289366016.0000 -
root_mean_squared_error: 282641.3125 - val_loss: 85907243008.0000 -
val_root_mean_squared_error: 293580.0625
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 768us/step - loss: 79729606656.0000 -
root_mean_squared_error: 281663.3438 - val_loss: 85857951744.0000 -
val_root_mean_squared_error: 293496.9062
Epoch 20/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 772us/step - loss: 79224406016.0000 -
root_mean_squared_error: 280777.6562 - val_loss: 85801091072.0000 -
val_root_mean_squared_error: 293396.7812
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 781us/step - loss: 78760648704.0000 -
root_mean_squared_error: 279961.1875 - val_loss: 85789966336.0000 -
```

```
val_root_mean_squared_error: 293373.7500
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 779us/step - loss: 78268792832.0000 -
root_mean_squared_error: 279095.1250 - val_loss: 85753765888.0000 -
val_root_mean_squared_error: 293312.1250
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 738us/step - loss: 77804822528.0000 -
root_mean_squared_error: 278277.9375 - val_loss: 85766938624.0000 -
val_root_mean_squared_error: 293336.5312
Epoch 24/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 718us/step - loss: 77374529536.0000 -
root_mean_squared_error: 277521.3750 - val_loss: 85846155264.0000 -
val_root_mean_squared_error: 293472.5000
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - loss: 76995330048.0000 -
root_mean_squared_error: 276849.9688 - val_loss: 85908021248.0000 -
val_root_mean_squared_error: 293578.5625
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 733us/step - loss: 76586098688.0000 -
root_mean_squared_error: 276124.2188 - val_loss: 86005833728.0000 -
val_root_mean_squared_error: 293745.4062
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 732us/step - loss: 76134195200.0000 -
root_mean_squared_error: 275314.7188 - val_loss: 86174334976.0000 -
val_root_mean_squared_error: 294029.5938
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 737us/step - loss: 75768864768.0000 -
root_mean_squared_error: 274661.3125 - val_loss: 86248554496.0000 -
val_root_mean_squared_error: 294157.4688
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 745us/step - loss: 75256504320.0000 -
root_mean_squared_error: 273739.0625 - val_loss: 86510075904.0000 -
val_root_mean_squared_error: 294601.1562
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 825us/step - loss: 74782769152.0000 -
root_mean_squared_error: 272871.3438 - val_loss: 86504357888.0000 -
val_root_mean_squared_error: 294593.3125
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 749us/step - loss: 74386202624.0000 -
root_mean_squared_error: 272154.3750 - val_loss: 86714761216.0000 -
val_root_mean_squared_error: 294949.3750
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 722us/step - loss: 73938526208.0000 -
root_mean_squared_error: 271348.1250 - val_loss: 87101284352.0000 -
val_root_mean_squared_error: 295603.1875
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 73472409600.0000 -
root_mean_squared_error: 270484.2500 - val_loss: 87231455232.0000 -
val_root_mean_squared_error: 295823.5625
```

```
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 738us/step - loss: 73069150208.0000 -
root_mean_squared_error: 269743.7812 - val_loss: 87693475840.0000 -
val_root_mean_squared_error: 296602.4688
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 762us/step - loss: 72694784000.0000 -
root_mean_squared_error: 269060.0000 - val_loss: 87813742592.0000 -
val_root_mean_squared_error: 296805.8125
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 744us/step - loss: 72291975168.0000 -
root_mean_squared_error: 268318.2500 - val_loss: 88211988480.0000 -
val_root_mean_squared_error: 297477.9688
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 725us/step - loss: 71943749632.0000 -
root_mean_squared_error: 267669.8438 - val_loss: 88346845184.0000 -
val_root_mean_squared_error: 297706.6562
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 729us/step - loss: 71437221888.0000 -
root_mean_squared_error: 266726.5000 - val_loss: 88669929472.0000 -
val_root_mean_squared_error: 298254.0000
Epoch 39/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 741us/step - loss: 71071539200.0000 -
root_mean_squared_error: 266047.7188 - val_loss: 88975392768.0000 -
val_root_mean_squared_error: 298760.0312
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 758us/step - loss: 70445203456.0000 -
root_mean_squared_error: 264867.0625 - val_loss: 88917819392.0000 -
val_root_mean_squared_error: 298679.5312
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 805us/step - loss: 70260187136.0000 -
root_mean_squared_error: 264535.1562 - val_loss: 89405693952.0000 -
val_root_mean_squared_error: 299495.0938
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 69962678272.0000 -
root_mean_squared_error: 263978.5625 - val_loss: 90257260544.0000 -
val_root_mean_squared_error: 300914.6875
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 743us/step - loss: 69313208320.0000 -
root_mean_squared_error: 262737.7188 - val_loss: 90154844160.0000 -
val_root_mean_squared_error: 300759.5625
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 69142396928.0000 -
root_mean_squared_error: 262422.2812 - val_loss: 90811621376.0000 -
val_root_mean_squared_error: 301846.8125
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 744us/step - loss: 68612980736.0000 -
root_mean_squared_error: 261400.3281 - val_loss: 90803757056.0000 -
val_root_mean_squared_error: 301850.3125
Epoch 46/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 781us/step - loss: 68574187520.0000 - root_mean_squared_error: 261344.5625 - val_loss: 91876483072.0000 - val_root_mean_squared_error: 303620.6562
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 776us/step - loss: 67904368640.0000 - root_mean_squared_error: 260051.2500 - val_loss: 91618492416.0000 - val_root_mean_squared_error: 303204.0625
Epoch 48/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 726us/step - loss: 67833274368.0000 - root_mean_squared_error: 259928.3750 - val_loss: 92323299328.0000 - val_root_mean_squared_error: 304360.3125
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 777us/step - loss: 67297513472.0000 - root_mean_squared_error: 258880.3750 - val_loss: 92258926592.0000 - val_root_mean_squared_error: 304264.9375
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 741us/step - loss: 67263270912.0000 - root_mean_squared_error: 258834.0625 - val_loss: 93399613440.0000 - val_root_mean_squared_error: 306130.5000
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 809us/step - loss: 66635739136.0000 - root_mean_squared_error: 257622.5781 - val_loss: 92948439040.0000 - val_root_mean_squared_error: 305394.7812
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 66602041344.0000 - root_mean_squared_error: 257565.5938 - val_loss: 94483357696.0000 - val_root_mean_squared_error: 307888.0000
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 741us/step - loss: 65961529344.0000 - root_mean_squared_error: 256316.1094 - val_loss: 94729961472.0000 - val_root_mean_squared_error: 308294.0000
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 65781932032.0000 - root_mean_squared_error: 255972.4688 - val_loss: 94842118144.0000 - val_root_mean_squared_error: 308471.6562
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 732us/step - loss: 65761869824.0000 - root_mean_squared_error: 255953.9375 - val_loss: 96110182400.0000 - val_root_mean_squared_error: 310512.0312
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 736us/step - loss: 65249529856.0000 - root_mean_squared_error: 254947.6094 - val_loss: 96700620800.0000 - val_root_mean_squared_error: 311476.4062
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 754us/step - loss: 64880226304.0000 - root_mean_squared_error: 254243.7344 - val_loss: 96725843968.0000 - val_root_mean_squared_error: 311522.3125
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 776us/step - loss: 64856137728.0000 -
```

```
root_mean_squared_error: 254200.7188 - val_loss: 98747277312.0000 -
val_root_mean_squared_error: 314747.7812
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 739us/step - loss: 64326197248.0000 -
root_mean_squared_error: 253154.1094 - val_loss: 98572312576.0000 -
val_root_mean_squared_error: 314477.4062
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 64050372608.0000 -
root_mean_squared_error: 252631.8281 - val_loss: 98061762560.0000 -
val_root_mean_squared_error: 313660.7812
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 744us/step - loss: 64025055232.0000 -
root_mean_squared_error: 252578.6719 - val_loss: 99884204032.0000 -
val_root_mean_squared_error: 316551.9375
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 724us/step - loss: 63393157120.0000 -
root_mean_squared_error: 251311.0781 - val_loss: 99707002880.0000 -
val_root_mean_squared_error: 316284.3750
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 727us/step - loss: 63137054720.0000 -
root_mean_squared_error: 250827.1406 - val_loss: 99252682752.0000 -
val_root_mean_squared_error: 315567.0938
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 732us/step - loss: 63284834304.0000 -
root_mean_squared_error: 251128.6562 - val_loss: 100655906816.0000 -
val_root_mean_squared_error: 317776.0312
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - loss: 62621315072.0000 -
root_mean_squared_error: 249790.7500 - val_loss: 101333188608.0000 -
val_root_mean_squared_error: 318858.8125
Epoch 66/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 739us/step - loss: 62597177344.0000 -
root_mean_squared_error: 249776.4844 - val_loss: 100930936832.0000 -
val_root_mean_squared_error: 318219.1875
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 741us/step - loss: 62700093440.0000 -
root_mean_squared_error: 249989.9844 - val_loss: 103541972992.0000 -
val_root_mean_squared_error: 322292.7188
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 766us/step - loss: 61785669632.0000 -
root_mean_squared_error: 248120.1250 - val_loss: 102157860864.0000 -
val_root_mean_squared_error: 320155.1875
Epoch 69/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 744us/step - loss: 62348029952.0000 -
root_mean_squared_error: 249307.4844 - val_loss: 103353458688.0000 -
val_root_mean_squared_error: 322013.9375
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 755us/step - loss: 61682040832.0000 -
root_mean_squared_error: 247951.0469 - val_loss: 103863468032.0000 -
```

```
val_root_mean_squared_error: 322800.0000
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 749us/step - loss: 61482569728.0000 -
root_mean_squared_error: 247541.3281 - val_loss: 103318372352.0000 -
val_root_mean_squared_error: 321960.4375
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 61794738176.0000 -
root_mean_squared_error: 248207.0000 - val_loss: 105090031616.0000 -
val_root_mean_squared_error: 324707.5312
Epoch 73/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 736us/step - loss: 61336223744.0000 -
root_mean_squared_error: 247278.2500 - val_loss: 104444903424.0000 -
val_root_mean_squared_error: 323706.9062
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 736us/step - loss: 61443166208.0000 -
root_mean_squared_error: 247492.3281 - val_loss: 107181072384.0000 -
val_root_mean_squared_error: 327909.4688
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 754us/step - loss: 60502978560.0000 -
root_mean_squared_error: 245561.3438 - val_loss: 104938004480.0000 -
val_root_mean_squared_error: 324482.4688
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 736us/step - loss: 61237850112.0000 -
root_mean_squared_error: 247096.8281 - val_loss: 106827857920.0000 -
val_root_mean_squared_error: 327386.3750
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 742us/step - loss: 60733505536.0000 -
root_mean_squared_error: 246054.7812 - val_loss: 106857070592.0000 -
val_root_mean_squared_error: 327421.4375
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 777us/step - loss: 60473479168.0000 -
root_mean_squared_error: 245510.6250 - val_loss: 106896433152.0000 -
val_root_mean_squared_error: 327494.0625
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 777us/step - loss: 60444798976.0000 -
root_mean_squared_error: 245484.7031 - val_loss: 105866452992.0000 -
val_root_mean_squared_error: 325910.0938
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 737us/step - loss: 60520607744.0000 -
root_mean_squared_error: 245645.4062 - val_loss: 108689137664.0000 -
val_root_mean_squared_error: 330213.4688
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 747us/step - loss: 59721547776.0000 -
root_mean_squared_error: 243987.4688 - val_loss: 106755383296.0000 -
val_root_mean_squared_error: 327282.1562
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 738us/step - loss: 60308832256.0000 -
root_mean_squared_error: 245217.3750 - val_loss: 108353011712.0000 -
val_root_mean_squared_error: 329721.9375
```

```
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 769us/step - loss: 59740897280.0000 -
root_mean_squared_error: 244053.5000 - val_loss: 108496502784.0000 -
val_root_mean_squared_error: 329932.7812
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 736us/step - loss: 59585011712.0000 -
root_mean_squared_error: 243719.7344 - val_loss: 109075660800.0000 -
val_root_mean_squared_error: 330829.0625
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 740us/step - loss: 59490750464.0000 -
root_mean_squared_error: 243553.6094 - val_loss: 110231658496.0000 -
val_root_mean_squared_error: 332558.8750
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 58619953152.0000 -
root_mean_squared_error: 241727.4844 - val_loss: 107459805184.0000 -
val_root_mean_squared_error: 328367.2812
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 734us/step - loss: 59440017408.0000 -
root_mean_squared_error: 243461.7344 - val_loss: 110273855488.0000 -
val_root_mean_squared_error: 332628.4375
Epoch 88/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 722us/step - loss: 58930388992.0000 -
root_mean_squared_error: 242400.8594 - val_loss: 110327857152.0000 -
val_root_mean_squared_error: 332712.7188
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 750us/step - loss: 58577305600.0000 -
root_mean_squared_error: 241660.4062 - val_loss: 110673248256.0000 -
val_root_mean_squared_error: 333233.3438
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 739us/step - loss: 58535731200.0000 -
root_mean_squared_error: 241594.8438 - val_loss: 111463604224.0000 -
val_root_mean_squared_error: 334417.8125
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 759us/step - loss: 58178260992.0000 -
root_mean_squared_error: 240837.9219 - val_loss: 111268831232.0000 -
val_root_mean_squared_error: 334130.7500
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 734us/step - loss: 58371858432.0000 -
root_mean_squared_error: 241271.5781 - val_loss: 114075893760.0000 -
val_root_mean_squared_error: 338318.7188
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 57817628672.0000 -
root_mean_squared_error: 240115.1406 - val_loss: 111850758144.0000 -
val_root_mean_squared_error: 334989.4375
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 728us/step - loss: 57870401536.0000 -
root_mean_squared_error: 240244.7656 - val_loss: 112424845312.0000 -
val_root_mean_squared_error: 335846.8750
Epoch 95/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 773us/step - loss: 57343311872.0000 -
root_mean_squared_error: 239120.2969 - val_loss: 112053936128.0000 -
val_root_mean_squared_error: 335306.1562
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 750us/step - loss: 57431949312.0000 -
root_mean_squared_error: 239340.5938 - val_loss: 114502402048.0000 -
val_root_mean_squared_error: 338948.9062
Epoch 97/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 754us/step - loss: 56665313280.0000 -
root_mean_squared_error: 237688.9531 - val_loss: 111713886208.0000 -
val_root_mean_squared_error: 334791.5312
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 741us/step - loss: 57629220864.0000 -
root_mean_squared_error: 239766.1250 - val_loss: 116694106112.0000 -
val_root_mean_squared_error: 342163.8750
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 737us/step - loss: 56539410432.0000 -
root_mean_squared_error: 237443.4219 - val_loss: 114404040704.0000 -
val_root_mean_squared_error: 338790.8750
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 771us/step - loss: 56746147840.0000 -
root_mean_squared_error: 237902.8438 - val_loss: 115659546624.0000 -
val_root_mean_squared_error: 340645.9688

<keras.src.callbacks.history.History at 0x23501f16650>

medium_nn = load_model('/tmp/ckpt/checkpoint.model.keras')
mse(medium_nn.predict(X_train), y_train, squared=False),
mse(medium_nn.predict(X_val), y_val, squared=False)

219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 553us/step
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 565us/step

(294185.518491572, 293312.1239026678)

history = medium_nn.fit(x=X_train,y=y_train, validation_data =
(X_val,y_val),callbacks = [cp], batch_size = 32, epochs = 100, verbose
= 1)
# Get training and test loss histories
training_loss = history.history['loss']
test_loss = history.history['val_loss']

# Create count of the number of epochs
epoch_count = range(1, len(training_loss) + 1)

Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 77804822528.0000 -
root_mean_squared_error: 278277.9375 - val_loss: 85766938624.0000 -
val_root_mean_squared_error: 293336.5312
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 718us/step - loss: 77374529536.0000 -
```

```
root_mean_squared_error: 277521.3750 - val_loss: 85846155264.0000 -
val_root_mean_squared_error: 293472.5000
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 724us/step - loss: 76995330048.0000 -
root_mean_squared_error: 276849.9688 - val_loss: 85908021248.0000 -
val_root_mean_squared_error: 293578.5625
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 76586098688.0000 -
root_mean_squared_error: 276124.2188 - val_loss: 86005833728.0000 -
val_root_mean_squared_error: 293745.4062
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 693us/step - loss: 76134195200.0000 -
root_mean_squared_error: 275314.7188 - val_loss: 86174334976.0000 -
val_root_mean_squared_error: 294029.5938
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 707us/step - loss: 75768864768.0000 -
root_mean_squared_error: 274661.3125 - val_loss: 86248554496.0000 -
val_root_mean_squared_error: 294157.4688
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 75256504320.0000 -
root_mean_squared_error: 273739.0625 - val_loss: 86510075904.0000 -
val_root_mean_squared_error: 294601.1562
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 687us/step - loss: 74782769152.0000 -
root_mean_squared_error: 272871.3438 - val_loss: 86504357888.0000 -
val_root_mean_squared_error: 294593.3125
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 678us/step - loss: 74386202624.0000 -
root_mean_squared_error: 272154.3750 - val_loss: 86714761216.0000 -
val_root_mean_squared_error: 294949.3750
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 73938526208.0000 -
root_mean_squared_error: 271348.1250 - val_loss: 87101284352.0000 -
val_root_mean_squared_error: 295603.1875
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 678us/step - loss: 73472409600.0000 -
root_mean_squared_error: 270484.2500 - val_loss: 87231455232.0000 -
val_root_mean_squared_error: 295823.5625
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 73069150208.0000 -
root_mean_squared_error: 269743.7812 - val_loss: 87693475840.0000 -
val_root_mean_squared_error: 296602.4688
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 682us/step - loss: 72694784000.0000 -
root_mean_squared_error: 269060.0000 - val_loss: 87813742592.0000 -
val_root_mean_squared_error: 296805.8125
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 686us/step - loss: 72291975168.0000 -
root_mean_squared_error: 268318.2500 - val_loss: 88211988480.0000 -
```

```
val_root_mean_squared_error: 297477.9688
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 71943749632.0000 -
root_mean_squared_error: 267669.8438 - val_loss: 88346845184.0000 -
val_root_mean_squared_error: 297706.6562
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 694us/step - loss: 71437221888.0000 -
root_mean_squared_error: 266726.5000 - val_loss: 88669929472.0000 -
val_root_mean_squared_error: 298254.0000
Epoch 17/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 706us/step - loss: 71071539200.0000 -
root_mean_squared_error: 266047.7188 - val_loss: 88975392768.0000 -
val_root_mean_squared_error: 298760.0312
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 70445203456.0000 -
root_mean_squared_error: 264867.0625 - val_loss: 88917819392.0000 -
val_root_mean_squared_error: 298679.5312
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 70260187136.0000 -
root_mean_squared_error: 264535.1562 - val_loss: 89405693952.0000 -
val_root_mean_squared_error: 299495.0938
Epoch 20/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 691us/step - loss: 69962678272.0000 -
root_mean_squared_error: 263978.5625 - val_loss: 90257260544.0000 -
val_root_mean_squared_error: 300914.6875
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 69313208320.0000 -
root_mean_squared_error: 262737.7188 - val_loss: 90154844160.0000 -
val_root_mean_squared_error: 300759.5625
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 693us/step - loss: 69142396928.0000 -
root_mean_squared_error: 262422.2812 - val_loss: 90811621376.0000 -
val_root_mean_squared_error: 301846.8125
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 68612980736.0000 -
root_mean_squared_error: 261400.3281 - val_loss: 90803757056.0000 -
val_root_mean_squared_error: 301850.3125
Epoch 24/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 689us/step - loss: 68574187520.0000 -
root_mean_squared_error: 261344.5625 - val_loss: 91876483072.0000 -
val_root_mean_squared_error: 303620.6562
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 685us/step - loss: 67904368640.0000 -
root_mean_squared_error: 260051.2500 - val_loss: 91618492416.0000 -
val_root_mean_squared_error: 303204.0625
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 67833274368.0000 -
root_mean_squared_error: 259928.3750 - val_loss: 92323299328.0000 -
val_root_mean_squared_error: 304360.3125
```

```
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 67297513472.0000 -
root_mean_squared_error: 258880.3750 - val_loss: 92258926592.0000 -
val_root_mean_squared_error: 304264.9375
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 67263270912.0000 -
root_mean_squared_error: 258834.0625 - val_loss: 93399613440.0000 -
val_root_mean_squared_error: 306130.5000
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 725us/step - loss: 66635739136.0000 -
root_mean_squared_error: 257622.5781 - val_loss: 92948439040.0000 -
val_root_mean_squared_error: 305394.7812
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 748us/step - loss: 66602041344.0000 -
root_mean_squared_error: 257565.5938 - val_loss: 94483357696.0000 -
val_root_mean_squared_error: 307888.0000
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 711us/step - loss: 65961529344.0000 -
root_mean_squared_error: 256316.1094 - val_loss: 94729961472.0000 -
val_root_mean_squared_error: 308294.0000
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 702us/step - loss: 65781932032.0000 -
root_mean_squared_error: 255972.4688 - val_loss: 94842118144.0000 -
val_root_mean_squared_error: 308471.6562
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 693us/step - loss: 65761869824.0000 -
root_mean_squared_error: 255953.9375 - val_loss: 96110182400.0000 -
val_root_mean_squared_error: 310512.0312
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 65249529856.0000 -
root_mean_squared_error: 254947.6094 - val_loss: 96700620800.0000 -
val_root_mean_squared_error: 311476.4062
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 709us/step - loss: 64880226304.0000 -
root_mean_squared_error: 254243.7344 - val_loss: 96725843968.0000 -
val_root_mean_squared_error: 311522.3125
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 706us/step - loss: 64856137728.0000 -
root_mean_squared_error: 254200.7188 - val_loss: 98747277312.0000 -
val_root_mean_squared_error: 314747.7812
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 64326197248.0000 -
root_mean_squared_error: 253154.1094 - val_loss: 98572312576.0000 -
val_root_mean_squared_error: 314477.4062
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 737us/step - loss: 64050372608.0000 -
root_mean_squared_error: 252631.8281 - val_loss: 98061762560.0000 -
val_root_mean_squared_error: 313660.7812
Epoch 39/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 717us/step - loss: 64025055232.0000 -
root_mean_squared_error: 252578.6719 - val_loss: 99884204032.0000 -
val_root_mean_squared_error: 316551.9375
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 63393157120.0000 -
root_mean_squared_error: 251311.0781 - val_loss: 99707002880.0000 -
val_root_mean_squared_error: 316284.3750
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 722us/step - loss: 63137054720.0000 -
root_mean_squared_error: 250827.1406 - val_loss: 99252682752.0000 -
val_root_mean_squared_error: 315567.0938
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 63284834304.0000 -
root_mean_squared_error: 251128.6562 - val_loss: 100655906816.0000 -
val_root_mean_squared_error: 317776.0312
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 701us/step - loss: 62621315072.0000 -
root_mean_squared_error: 249790.7500 - val_loss: 101333188608.0000 -
val_root_mean_squared_error: 318858.8125
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 728us/step - loss: 62597177344.0000 -
root_mean_squared_error: 249776.4844 - val_loss: 100930936832.0000 -
val_root_mean_squared_error: 318219.1875
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 707us/step - loss: 62700093440.0000 -
root_mean_squared_error: 249989.9844 - val_loss: 103541972992.0000 -
val_root_mean_squared_error: 322292.7188
Epoch 46/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 61785669632.0000 -
root_mean_squared_error: 248120.1250 - val_loss: 102157860864.0000 -
val_root_mean_squared_error: 320155.1875
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 714us/step - loss: 62348029952.0000 -
root_mean_squared_error: 249307.4844 - val_loss: 103353458688.0000 -
val_root_mean_squared_error: 322013.9375
Epoch 48/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 61682040832.0000 -
root_mean_squared_error: 247951.0469 - val_loss: 103863468032.0000 -
val_root_mean_squared_error: 322800.0000
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 702us/step - loss: 61482569728.0000 -
root_mean_squared_error: 247541.3281 - val_loss: 103318372352.0000 -
val_root_mean_squared_error: 321960.4375
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 733us/step - loss: 61794738176.0000 -
root_mean_squared_error: 248207.0000 - val_loss: 105090031616.0000 -
val_root_mean_squared_error: 324707.5312
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 710us/step - loss: 61336223744.0000 -
```

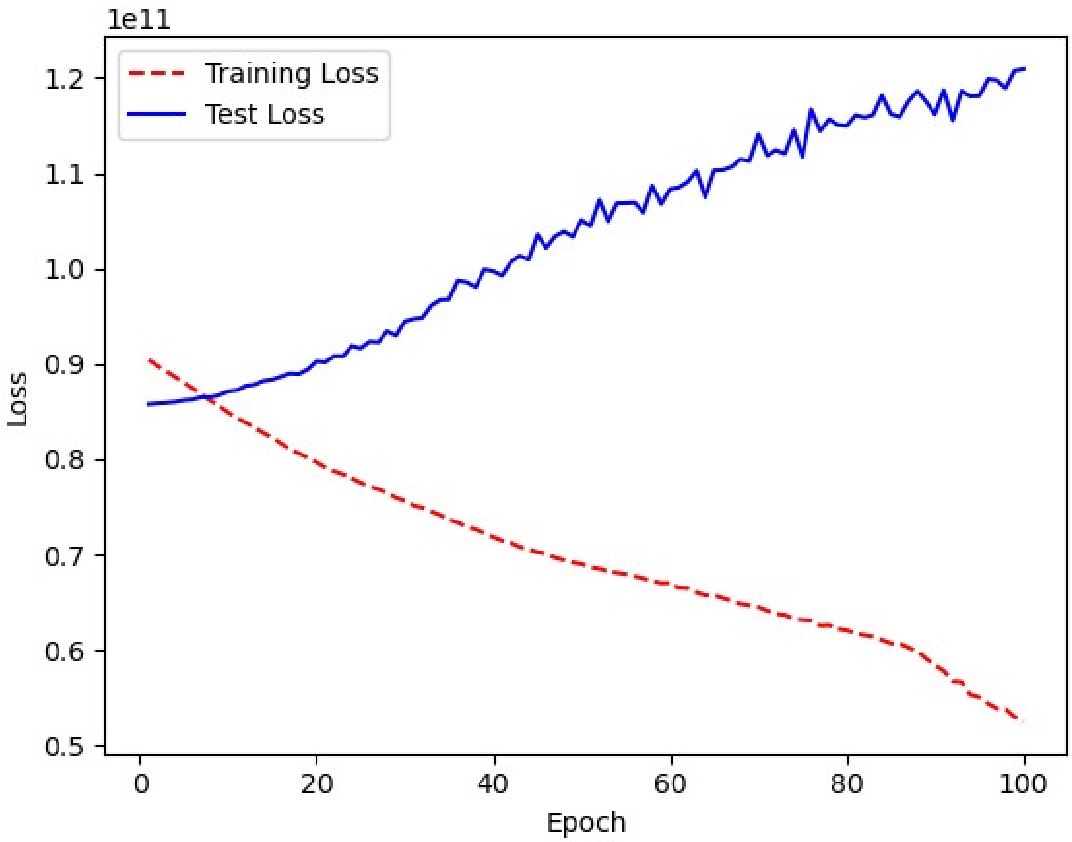
```
root_mean_squared_error: 247278.2500 - val_loss: 104444903424.0000 -
val_root_mean_squared_error: 323706.9062
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 61443166208.0000 -
root_mean_squared_error: 247492.3281 - val_loss: 107181072384.0000 -
val_root_mean_squared_error: 327909.4688
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 706us/step - loss: 60502978560.0000 -
root_mean_squared_error: 245561.3438 - val_loss: 104938004480.0000 -
val_root_mean_squared_error: 324482.4688
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 713us/step - loss: 61237850112.0000 -
root_mean_squared_error: 247096.8281 - val_loss: 106827857920.0000 -
val_root_mean_squared_error: 327386.3750
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 732us/step - loss: 60733505536.0000 -
root_mean_squared_error: 246054.7812 - val_loss: 106857070592.0000 -
val_root_mean_squared_error: 327421.4375
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 724us/step - loss: 60473479168.0000 -
root_mean_squared_error: 245510.6250 - val_loss: 106896433152.0000 -
val_root_mean_squared_error: 327494.0625
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 60444798976.0000 -
root_mean_squared_error: 245484.7031 - val_loss: 105866452992.0000 -
val_root_mean_squared_error: 325910.0938
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 60520607744.0000 -
root_mean_squared_error: 245645.4062 - val_loss: 108689137664.0000 -
val_root_mean_squared_error: 330213.4688
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 59721547776.0000 -
root_mean_squared_error: 243987.4688 - val_loss: 106755383296.0000 -
val_root_mean_squared_error: 327282.1562
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 892us/step - loss: 60308832256.0000 -
root_mean_squared_error: 245217.3750 - val_loss: 108353011712.0000 -
val_root_mean_squared_error: 329721.9375
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 956us/step - loss: 59740897280.0000 -
root_mean_squared_error: 244053.5000 - val_loss: 108496502784.0000 -
val_root_mean_squared_error: 329932.7812
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 832us/step - loss: 59585011712.0000 -
root_mean_squared_error: 243719.7344 - val_loss: 109075660800.0000 -
val_root_mean_squared_error: 330829.0625
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 841us/step - loss: 59490750464.0000 -
root_mean_squared_error: 243553.6094 - val_loss: 110231658496.0000 -
```

```
val_root_mean_squared_error: 332558.8750
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 58619953152.0000 -
root_mean_squared_error: 241727.4844 - val_loss: 107459805184.0000 -
val_root_mean_squared_error: 328367.2812
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 796us/step - loss: 59440017408.0000 -
root_mean_squared_error: 243461.7344 - val_loss: 110273855488.0000 -
val_root_mean_squared_error: 332628.4375
Epoch 66/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 58930388992.0000 -
root_mean_squared_error: 242400.8594 - val_loss: 110327857152.0000 -
val_root_mean_squared_error: 332712.7188
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 764us/step - loss: 58577305600.0000 -
root_mean_squared_error: 241660.4062 - val_loss: 110673248256.0000 -
val_root_mean_squared_error: 333233.3438
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 746us/step - loss: 58535731200.0000 -
root_mean_squared_error: 241594.8438 - val_loss: 111463604224.0000 -
val_root_mean_squared_error: 334417.8125
Epoch 69/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 745us/step - loss: 58178260992.0000 -
root_mean_squared_error: 240837.9219 - val_loss: 111268831232.0000 -
val_root_mean_squared_error: 334130.7500
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 775us/step - loss: 58371858432.0000 -
root_mean_squared_error: 241271.5781 - val_loss: 114075893760.0000 -
val_root_mean_squared_error: 338318.7188
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 821us/step - loss: 57817628672.0000 -
root_mean_squared_error: 240115.1406 - val_loss: 111850758144.0000 -
val_root_mean_squared_error: 334989.4375
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 794us/step - loss: 57870401536.0000 -
root_mean_squared_error: 240244.7656 - val_loss: 112424845312.0000 -
val_root_mean_squared_error: 335846.8750
Epoch 73/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 701us/step - loss: 57343311872.0000 -
root_mean_squared_error: 239120.2969 - val_loss: 112053936128.0000 -
val_root_mean_squared_error: 335306.1562
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 694us/step - loss: 57431949312.0000 -
root_mean_squared_error: 239340.5938 - val_loss: 114502402048.0000 -
val_root_mean_squared_error: 338948.9062
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step - loss: 56665313280.0000 -
root_mean_squared_error: 237688.9531 - val_loss: 111713886208.0000 -
val_root_mean_squared_error: 334791.5312
```

```
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 690us/step - loss: 57629220864.0000 -
root_mean_squared_error: 239766.1250 - val_loss: 116694106112.0000 -
val_root_mean_squared_error: 342163.8750
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 710us/step - loss: 56539410432.0000 -
root_mean_squared_error: 237443.4219 - val_loss: 114404040704.0000 -
val_root_mean_squared_error: 338790.8750
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 704us/step - loss: 56746147840.0000 -
root_mean_squared_error: 237902.8438 - val_loss: 115659546624.0000 -
val_root_mean_squared_error: 340645.9688
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 56588603392.0000 -
root_mean_squared_error: 237566.3438 - val_loss: 115079340032.0000 -
val_root_mean_squared_error: 339785.1250
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 705us/step - loss: 56558129152.0000 -
root_mean_squared_error: 237520.6250 - val_loss: 114956820480.0000 -
val_root_mean_squared_error: 339603.9688
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 692us/step - loss: 56085528576.0000 -
root_mean_squared_error: 236497.4844 - val_loss: 116085080064.0000 -
val_root_mean_squared_error: 341246.5625
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 701us/step - loss: 56139329536.0000 -
root_mean_squared_error: 236640.5625 - val_loss: 115853918208.0000 -
val_root_mean_squared_error: 340935.2188
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 702us/step - loss: 55874101248.0000 -
root_mean_squared_error: 236060.3594 - val_loss: 116094050304.0000 -
val_root_mean_squared_error: 341270.3438
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 715us/step - loss: 55828697088.0000 -
root_mean_squared_error: 235986.2188 - val_loss: 118141140992.0000 -
val_root_mean_squared_error: 344284.6250
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 707us/step - loss: 55117234176.0000 -
root_mean_squared_error: 234442.4375 - val_loss: 116192641024.0000 -
val_root_mean_squared_error: 341428.4375
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 701us/step - loss: 55383441408.0000 -
root_mean_squared_error: 235046.5781 - val_loss: 115908435968.0000 -
val_root_mean_squared_error: 341013.0938
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 698us/step - loss: 55109513216.0000 -
root_mean_squared_error: 234448.5781 - val_loss: 117522415616.0000 -
val_root_mean_squared_error: 343373.5312
Epoch 88/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 55006556160.0000 -
root_mean_squared_error: 234248.0469 - val_loss: 118617006080.0000 -
val_root_mean_squared_error: 344972.2500
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 53800673280.0000 -
root_mean_squared_error: 231618.9219 - val_loss: 117452341248.0000 -
val_root_mean_squared_error: 343285.5625
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 714us/step - loss: 53301657600.0000 -
root_mean_squared_error: 230555.0000 - val_loss: 116179435520.0000 -
val_root_mean_squared_error: 341434.7500
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 698us/step - loss: 53019709440.0000 -
root_mean_squared_error: 229952.6250 - val_loss: 118703521792.0000 -
val_root_mean_squared_error: 345126.5625
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 51649617920.0000 -
root_mean_squared_error: 226919.2500 - val_loss: 115557957632.0000 -
val_root_mean_squared_error: 340546.2500
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 706us/step - loss: 52108963840.0000 -
root_mean_squared_error: 227978.4375 - val_loss: 118645710848.0000 -
val_root_mean_squared_error: 345058.7812
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 697us/step - loss: 50475507712.0000 -
root_mean_squared_error: 224344.3750 - val_loss: 118061580288.0000 -
val_root_mean_squared_error: 344227.3125
Epoch 95/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 699us/step - loss: 50524565504.0000 -
root_mean_squared_error: 224486.6719 - val_loss: 118092726272.0000 -
val_root_mean_squared_error: 344273.4062
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 729us/step - loss: 49840140288.0000 -
root_mean_squared_error: 222929.4219 - val_loss: 119877058560.0000 -
val_root_mean_squared_error: 346857.1875
Epoch 97/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 893us/step - loss: 49027014656.0000 -
root_mean_squared_error: 221086.7188 - val_loss: 119743782912.0000 -
val_root_mean_squared_error: 346691.6562
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 731us/step - loss: 49662668800.0000 -
root_mean_squared_error: 222576.2344 - val_loss: 118939672576.0000 -
val_root_mean_squared_error: 345524.9688
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 695us/step - loss: 48787632128.0000 -
root_mean_squared_error: 220572.0625 - val_loss: 120734982144.0000 -
val_root_mean_squared_error: 348121.5000
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step - loss: 48401944576.0000 -
```

```
root_mean_squared_error: 219712.9844 - val_loss: 120918630400.0000 -
val_root_mean_squared_error: 348383.7500

# Visualize loss history for the medium neural network
plt.plot(epoch_count, training_loss, 'r--')
plt.plot(epoch_count, test_loss, 'b-')
plt.legend(['Training Loss', 'Test Loss'])
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show();
```



Large Neural Network

```
large_nn = Sequential()
large_nn.add(InputLayer((818,)))
large_nn.add(Dense(256, 'relu'))
large_nn.add(Dense(128, 'relu'))
large_nn.add(Dense(64, 'relu'))
large_nn.add(Dense(32, 'relu'))
large_nn.add(Dense(1, 'linear'))

opt = Adam(learning_rate=.1)
cp = ModelCheckpoint('/tmp/ckpt/checkpoint.model.keras',
```

```
save_best_only=True)
large_nn.compile(optimizer=opt, loss='mse',
metrics=[RootMeanSquaredError()])
large_nn.fit(x=X_train, y=y_train, validation_data=(X_val, y_val),
callbacks=[cp], epochs=100)
```

```
Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 330063118336.0000 -
root_mean_squared_error: 555799.7500 - val_loss: 137448439808.0000 -
val_root_mean_squared_error: 371131.8750
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 128818053120.0000 -
root_mean_squared_error: 358146.4375 - val_loss: 155315060736.0000 -
val_root_mean_squared_error: 394453.4375
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 113000226816.0000 -
root_mean_squared_error: 335786.0312 - val_loss: 82040438784.0000 -
val_root_mean_squared_error: 286788.6250
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 85376688128.0000 -
root_mean_squared_error: 291224.7812 - val_loss: 75709038592.0000 -
val_root_mean_squared_error: 275395.7812
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 75196465152.0000 -
root_mean_squared_error: 273402.7188 - val_loss: 71638507520.0000 -
val_root_mean_squared_error: 267929.2188
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 69181399040.0000 -
root_mean_squared_error: 262210.8438 - val_loss: 72446885888.0000 -
val_root_mean_squared_error: 269393.1562
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 66117824512.0000 -
root_mean_squared_error: 256594.7031 - val_loss: 70861070336.0000 -
val_root_mean_squared_error: 266446.6562
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 60182953984.0000 -
root_mean_squared_error: 244820.7969 - val_loss: 72503050240.0000 -
val_root_mean_squared_error: 269500.3125
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 56248946688.0000 -
root_mean_squared_error: 236554.3594 - val_loss: 75055603712.0000 -
val_root_mean_squared_error: 274154.5312
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 57550491648.0000 -
root_mean_squared_error: 238855.4844 - val_loss: 87095672832.0000 -
val_root_mean_squared_error: 295496.0625
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54508470272.0000 -
root_mean_squared_error: 233240.6719 - val_loss: 88485134336.0000 -
val_root_mean_squared_error: 297851.4688
```

```
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 57863704576.0000 -
root_mean_squared_error: 240244.4375 - val_loss: 78806097920.0000 -
val_root_mean_squared_error: 280925.2188
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54859141120.0000 -
root_mean_squared_error: 233322.0000 - val_loss: 85910937600.0000 -
val_root_mean_squared_error: 293060.0938
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 55928815616.0000 -
root_mean_squared_error: 235414.4219 - val_loss: 84900413440.0000 -
val_root_mean_squared_error: 291446.5312
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54969135104.0000 -
root_mean_squared_error: 233987.7344 - val_loss: 78206984192.0000 -
val_root_mean_squared_error: 279851.6875
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 46817861632.0000 -
root_mean_squared_error: 216096.7344 - val_loss: 76530114560.0000 -
val_root_mean_squared_error: 276417.0000
Epoch 17/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 44919775232.0000 -
root_mean_squared_error: 211306.2812 - val_loss: 75300225024.0000 -
val_root_mean_squared_error: 274005.2500
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40827113472.0000 -
root_mean_squared_error: 201399.6875 - val_loss: 77106053120.0000 -
val_root_mean_squared_error: 277711.5000
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 37881552896.0000 -
root_mean_squared_error: 194237.6250 - val_loss: 83403063296.0000 -
val_root_mean_squared_error: 288974.2188
Epoch 20/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42830307328.0000 -
root_mean_squared_error: 206431.2969 - val_loss: 98612011008.0000 -
val_root_mean_squared_error: 314103.5000
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40138092544.0000 -
root_mean_squared_error: 200080.4688 - val_loss: 105385492480.0000 -
val_root_mean_squared_error: 324810.6250
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40826982400.0000 -
root_mean_squared_error: 201917.4375 - val_loss: 106365247488.0000 -
val_root_mean_squared_error: 326407.8750
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40224907264.0000 -
root_mean_squared_error: 200267.9844 - val_loss: 89823551488.0000 -
val_root_mean_squared_error: 299647.2500
Epoch 24/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 38062686208.0000 -
root_mean_squared_error: 194415.3906 - val_loss: 92718833664.0000 -
val_root_mean_squared_error: 304383.5000
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 35783782400.0000 -
root_mean_squared_error: 188533.5938 - val_loss: 95432998912.0000 -
val_root_mean_squared_error: 308884.2500
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33038594048.0000 -
root_mean_squared_error: 181319.5781 - val_loss: 113901314048.0000 -
val_root_mean_squared_error: 337240.3125
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33947709440.0000 -
root_mean_squared_error: 183724.8125 - val_loss: 145409523712.0000 -
val_root_mean_squared_error: 380944.8125
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 31714150400.0000 -
root_mean_squared_error: 177825.3438 - val_loss: 172095160320.0000 -
val_root_mean_squared_error: 413867.0312
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 34599313408.0000 -
root_mean_squared_error: 185224.2344 - val_loss: 177837375488.0000 -
val_root_mean_squared_error: 420545.5938
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33633409024.0000 -
root_mean_squared_error: 183036.5781 - val_loss: 181393293312.0000 -
val_root_mean_squared_error: 424675.2812
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 37721427968.0000 -
root_mean_squared_error: 193557.3906 - val_loss: 171955765248.0000 -
val_root_mean_squared_error: 413625.6875
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 44675092480.0000 -
root_mean_squared_error: 210696.6406 - val_loss: 132505387008.0000 -
val_root_mean_squared_error: 362950.1562
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 52217159680.0000 -
root_mean_squared_error: 227820.6406 - val_loss: 122312409088.0000 -
val_root_mean_squared_error: 348769.5938
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 59259883520.0000 -
root_mean_squared_error: 242556.9062 - val_loss: 101818892288.0000 -
val_root_mean_squared_error: 318966.3438
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 47973474304.0000 -
root_mean_squared_error: 217848.9062 - val_loss: 94834302976.0000 -
val_root_mean_squared_error: 308160.0938
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 46558228480.0000 -
```

```
root_mean_squared_error: 214984.2344 - val_loss: 121336864768.0000 -
val_root_mean_squared_error: 348735.0625
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 47827804160.0000 -
root_mean_squared_error: 217569.0156 - val_loss: 119992819712.0000 -
val_root_mean_squared_error: 346598.3750
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42791772160.0000 -
root_mean_squared_error: 205805.7500 - val_loss: 111535497216.0000 -
val_root_mean_squared_error: 333928.3125
Epoch 39/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 80407166976.0000 -
root_mean_squared_error: 275982.7500 - val_loss: 94003757056.0000 -
val_root_mean_squared_error: 306557.1250
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 32372895744.0000 -
root_mean_squared_error: 179877.3594 - val_loss: 85094948864.0000 -
val_root_mean_squared_error: 291727.2500
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23048769536.0000 -
root_mean_squared_error: 151735.9688 - val_loss: 94367162368.0000 -
val_root_mean_squared_error: 307271.3125
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20433405952.0000 -
root_mean_squared_error: 142883.4688 - val_loss: 90934247424.0000 -
val_root_mean_squared_error: 301394.8125
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18574284800.0000 -
root_mean_squared_error: 136155.7969 - val_loss: 91156594688.0000 -
val_root_mean_squared_error: 302071.0625
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 16524505088.0000 -
root_mean_squared_error: 128364.1875 - val_loss: 93396566016.0000 -
val_root_mean_squared_error: 305340.1250
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 16133852160.0000 -
root_mean_squared_error: 126762.7891 - val_loss: 91973918720.0000 -
val_root_mean_squared_error: 303128.6875
Epoch 46/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 15483287552.0000 -
root_mean_squared_error: 124266.2500 - val_loss: 95709446144.0000 -
val_root_mean_squared_error: 309089.1562
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 14578180096.0000 -
root_mean_squared_error: 120592.1953 - val_loss: 98561654784.0000 -
val_root_mean_squared_error: 313707.5625
Epoch 48/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 15841553408.0000 -
root_mean_squared_error: 125572.2969 - val_loss: 95438798848.0000 -
```

```
val_root_mean_squared_error: 308420.0938
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 15873474560.0000 -
root_mean_squared_error: 125656.8359 - val_loss: 97700126720.0000 -
val_root_mean_squared_error: 312695.1875
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 15610659840.0000 -
root_mean_squared_error: 124569.8828 - val_loss: 96966254592.0000 -
val_root_mean_squared_error: 311236.5625
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 16079316992.0000 -
root_mean_squared_error: 126249.9844 - val_loss: 104021499904.0000 -
val_root_mean_squared_error: 322392.3125
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 17827260416.0000 -
root_mean_squared_error: 132712.4688 - val_loss: 101800919040.0000 -
val_root_mean_squared_error: 319029.3125
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20882669568.0000 -
root_mean_squared_error: 143205.0781 - val_loss: 120419500032.0000 -
val_root_mean_squared_error: 347312.8438
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 30739800064.0000 -
root_mean_squared_error: 173888.9531 - val_loss: 130490998784.0000 -
val_root_mean_squared_error: 361702.5938
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42872123392.0000 -
root_mean_squared_error: 206784.1406 - val_loss: 114060263424.0000 -
val_root_mean_squared_error: 338098.6562
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 38375317504.0000 -
root_mean_squared_error: 195701.6562 - val_loss: 95601483776.0000 -
val_root_mean_squared_error: 309518.6562
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42306641920.0000 -
root_mean_squared_error: 204829.3125 - val_loss: 93370638336.0000 -
val_root_mean_squared_error: 305640.6562
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 27306375168.0000 -
root_mean_squared_error: 164934.3438 - val_loss: 93808918528.0000 -
val_root_mean_squared_error: 306423.0938
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 22754328576.0000 -
root_mean_squared_error: 150595.6719 - val_loss: 94127423488.0000 -
val_root_mean_squared_error: 306938.9688
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20890413056.0000 -
root_mean_squared_error: 144292.1875 - val_loss: 97291083776.0000 -
val_root_mean_squared_error: 311891.8750
```

```
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 22408595456.0000 -
root_mean_squared_error: 149453.2344 - val_loss: 106768277504.0000 -
val_root_mean_squared_error: 326939.5312
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 19599093760.0000 -
root_mean_squared_error: 139715.9531 - val_loss: 111245041664.0000 -
val_root_mean_squared_error: 333626.8125
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18367580160.0000 -
root_mean_squared_error: 135424.0000 - val_loss: 121050177536.0000 -
val_root_mean_squared_error: 347828.4688
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18238945280.0000 -
root_mean_squared_error: 134862.6094 - val_loss: 131129458688.0000 -
val_root_mean_squared_error: 361863.1875
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20369037312.0000 -
root_mean_squared_error: 142608.6094 - val_loss: 148322336768.0000 -
val_root_mean_squared_error: 384279.5938
Epoch 66/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 19444164608.0000 -
root_mean_squared_error: 139288.0000 - val_loss: 137841344512.0000 -
val_root_mean_squared_error: 370579.6562
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 17451395072.0000 -
root_mean_squared_error: 132008.3125 - val_loss: 162856386560.0000 -
val_root_mean_squared_error: 403177.3125
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 17653540864.0000 -
root_mean_squared_error: 132725.7500 - val_loss: 162548957184.0000 -
val_root_mean_squared_error: 402708.1875
Epoch 69/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18823653376.0000 -
root_mean_squared_error: 136867.7812 - val_loss: 180104101888.0000 -
val_root_mean_squared_error: 423943.4688
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 24401852416.0000 -
root_mean_squared_error: 155662.4219 - val_loss: 157815701504.0000 -
val_root_mean_squared_error: 396651.2500
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20369852416.0000 -
root_mean_squared_error: 142530.0625 - val_loss: 137976561664.0000 -
val_root_mean_squared_error: 371148.9375
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 22993967104.0000 -
root_mean_squared_error: 151489.5938 - val_loss: 180584136704.0000 -
val_root_mean_squared_error: 424157.6875
Epoch 73/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 27006674944.0000 -
root_mean_squared_error: 163826.7812 - val_loss: 234545561600.0000 -
val_root_mean_squared_error: 483315.5938
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 31787866112.0000 -
root_mean_squared_error: 177627.2344 - val_loss: 190791352320.0000 -
val_root_mean_squared_error: 436376.6562
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42123214848.0000 -
root_mean_squared_error: 204105.0312 - val_loss: 123978997760.0000 -
val_root_mean_squared_error: 352136.6875
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 51205672960.0000 -
root_mean_squared_error: 224604.6250 - val_loss: 96891092992.0000 -
val_root_mean_squared_error: 311464.4375
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 45626081280.0000 -
root_mean_squared_error: 212370.1562 - val_loss: 108143779840.0000 -
val_root_mean_squared_error: 328684.8125
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 35210838016.0000 -
root_mean_squared_error: 186438.5781 - val_loss: 119187193856.0000 -
val_root_mean_squared_error: 345078.5625
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 29590577152.0000 -
root_mean_squared_error: 171031.4531 - val_loss: 119845355520.0000 -
val_root_mean_squared_error: 346098.3438
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 27594123264.0000 -
root_mean_squared_error: 165440.5938 - val_loss: 113804836864.0000 -
val_root_mean_squared_error: 337146.9062
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23912464384.0000 -
root_mean_squared_error: 153876.4375 - val_loss: 110573101056.0000 -
val_root_mean_squared_error: 332147.6875
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23942627328.0000 -
root_mean_squared_error: 154439.6250 - val_loss: 96609951744.0000 -
val_root_mean_squared_error: 310669.8125
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 21074219008.0000 -
root_mean_squared_error: 144584.0938 - val_loss: 95720562688.0000 -
val_root_mean_squared_error: 309226.1562
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 22246963200.0000 -
root_mean_squared_error: 148109.9375 - val_loss: 96003309568.0000 -
val_root_mean_squared_error: 310108.5625
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 19527573504.0000 -
```

```
root_mean_squared_error: 139057.2500 - val_loss: 97744134144.0000 -
val_root_mean_squared_error: 312685.7500
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18740848640.0000 -
root_mean_squared_error: 136130.9688 - val_loss: 99798073344.0000 -
val_root_mean_squared_error: 315889.8125
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 25471275008.0000 -
root_mean_squared_error: 156467.3281 - val_loss: 97396056064.0000 -
val_root_mean_squared_error: 312238.3125
Epoch 88/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20355934208.0000 -
root_mean_squared_error: 141939.5156 - val_loss: 99116351488.0000 -
val_root_mean_squared_error: 315098.4688
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 22505570304.0000 -
root_mean_squared_error: 149309.4219 - val_loss: 109052796928.0000 -
val_root_mean_squared_error: 330194.6875
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 21992380416.0000 -
root_mean_squared_error: 147933.7500 - val_loss: 101889622016.0000 -
val_root_mean_squared_error: 319528.2500
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 24229808128.0000 -
root_mean_squared_error: 155263.1875 - val_loss: 108094996480.0000 -
val_root_mean_squared_error: 329054.0312
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23525470208.0000 -
root_mean_squared_error: 152940.8438 - val_loss: 104201256960.0000 -
val_root_mean_squared_error: 323173.9688
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 28334966784.0000 -
root_mean_squared_error: 168003.7031 - val_loss: 125490495488.0000 -
val_root_mean_squared_error: 354407.5938
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 27497064448.0000 -
root_mean_squared_error: 165417.8125 - val_loss: 140326780928.0000 -
val_root_mean_squared_error: 374575.2500
Epoch 95/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 27553630208.0000 -
root_mean_squared_error: 165551.8750 - val_loss: 195534422016.0000 -
val_root_mean_squared_error: 441762.4688
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 28784975872.0000 -
root_mean_squared_error: 169313.7344 - val_loss: 186155188224.0000 -
val_root_mean_squared_error: 431174.4688
Epoch 97/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 76350316544.0000 -
root_mean_squared_error: 267227.3750 - val_loss: 117948645376.0000 -
```

```
val_root_mean_squared_error: 343607.2812
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 36313653248.0000 -
root_mean_squared_error: 190397.5625 - val_loss: 99248136192.0000 -
val_root_mean_squared_error: 315244.1875
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23255791616.0000 -
root_mean_squared_error: 152318.0625 - val_loss: 94537228288.0000 -
val_root_mean_squared_error: 307671.5938
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18679957504.0000 -
root_mean_squared_error: 136516.8125 - val_loss: 93514661888.0000 -
val_root_mean_squared_error: 305891.8125

<keras.src.callbacks.history.History at 0x2350d0e8950>
```

```
large_nn = load_model('/tmp/ckpt/checkpoint.model.keras')
mse(large_nn.predict(X_train), y_train, squared=False),
mse(large_nn.predict(X_val), y_val, squared=False)
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 784us/step
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 700us/step

(240594.6771277807, 266446.64886823954)
```

```
history = large_nn.fit(x=X_train,y=y_train, validation_data =
(X_val,y_val),callbacks = [cp], batch_size = 32, epochs = 100, verbose
= 1)
# Get training and test loss histories
training_loss = history.history['loss']
test_loss = history.history['val_loss']

# Create count of the number of epochs
epoch_count = range(1, len(training_loss) + 1)
```

```
Epoch 1/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 60182953984.0000 -
root_mean_squared_error: 244820.7969 - val_loss: 72503050240.0000 -
val_root_mean_squared_error: 269500.3125
Epoch 2/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 56248946688.0000 -
root_mean_squared_error: 236554.3594 - val_loss: 75055603712.0000 -
val_root_mean_squared_error: 274154.5312
Epoch 3/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 57550491648.0000 -
root_mean_squared_error: 238855.4844 - val_loss: 87095672832.0000 -
val_root_mean_squared_error: 295496.0625
Epoch 4/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54508470272.0000 -
root_mean_squared_error: 233240.6719 - val_loss: 88485134336.0000 -
val_root_mean_squared_error: 297851.4688
```

```
Epoch 5/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 57863704576.0000 -
root_mean_squared_error: 240244.4375 - val_loss: 78806097920.0000 -
val_root_mean_squared_error: 280925.2188
Epoch 6/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54859141120.0000 -
root_mean_squared_error: 233322.0000 - val_loss: 85910937600.0000 -
val_root_mean_squared_error: 293060.0938
Epoch 7/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 55928815616.0000 -
root_mean_squared_error: 235414.4219 - val_loss: 84900413440.0000 -
val_root_mean_squared_error: 291446.5312
Epoch 8/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 54969135104.0000 -
root_mean_squared_error: 233987.7344 - val_loss: 78206984192.0000 -
val_root_mean_squared_error: 279851.6875
Epoch 9/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 46817861632.0000 -
root_mean_squared_error: 216096.7344 - val_loss: 76530114560.0000 -
val_root_mean_squared_error: 276417.0000
Epoch 10/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 44919775232.0000 -
root_mean_squared_error: 211306.2812 - val_loss: 75300225024.0000 -
val_root_mean_squared_error: 274005.2500
Epoch 11/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40827113472.0000 -
root_mean_squared_error: 201399.6875 - val_loss: 77106053120.0000 -
val_root_mean_squared_error: 277711.5000
Epoch 12/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 37881552896.0000 -
root_mean_squared_error: 194237.6250 - val_loss: 83403063296.0000 -
val_root_mean_squared_error: 288974.2188
Epoch 13/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42830307328.0000 -
root_mean_squared_error: 206431.2969 - val_loss: 98612011008.0000 -
val_root_mean_squared_error: 314103.5000
Epoch 14/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40138092544.0000 -
root_mean_squared_error: 200080.4688 - val_loss: 105385492480.0000 -
val_root_mean_squared_error: 324810.6250
Epoch 15/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40826982400.0000 -
root_mean_squared_error: 201917.4375 - val_loss: 106365247488.0000 -
val_root_mean_squared_error: 326407.8750
Epoch 16/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 40224907264.0000 -
root_mean_squared_error: 200267.9844 - val_loss: 89823551488.0000 -
val_root_mean_squared_error: 299647.2500
Epoch 17/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 38062686208.0000 -
root_mean_squared_error: 194415.3906 - val_loss: 92718833664.0000 -
val_root_mean_squared_error: 304383.5000
Epoch 18/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 35783782400.0000 -
root_mean_squared_error: 188533.5938 - val_loss: 95432998912.0000 -
val_root_mean_squared_error: 308884.2500
Epoch 19/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33038594048.0000 -
root_mean_squared_error: 181319.5781 - val_loss: 113901314048.0000 -
val_root_mean_squared_error: 337240.3125
Epoch 20/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33947709440.0000 -
root_mean_squared_error: 183724.8125 - val_loss: 145409523712.0000 -
val_root_mean_squared_error: 380944.8125
Epoch 21/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 31714150400.0000 -
root_mean_squared_error: 177825.3438 - val_loss: 172095160320.0000 -
val_root_mean_squared_error: 413867.0312
Epoch 22/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 34599313408.0000 -
root_mean_squared_error: 185224.2344 - val_loss: 177837375488.0000 -
val_root_mean_squared_error: 420545.5938
Epoch 23/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 33633409024.0000 -
root_mean_squared_error: 183036.5781 - val_loss: 181393293312.0000 -
val_root_mean_squared_error: 424675.2812
Epoch 24/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 37721427968.0000 -
root_mean_squared_error: 193557.3906 - val_loss: 171955765248.0000 -
val_root_mean_squared_error: 413625.6875
Epoch 25/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 44675092480.0000 -
root_mean_squared_error: 210696.6406 - val_loss: 132505387008.0000 -
val_root_mean_squared_error: 362950.1562
Epoch 26/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 52217159680.0000 -
root_mean_squared_error: 227820.6406 - val_loss: 122312409088.0000 -
val_root_mean_squared_error: 348769.5938
Epoch 27/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 59259883520.0000 -
root_mean_squared_error: 242556.9062 - val_loss: 101818892288.0000 -
val_root_mean_squared_error: 318966.3438
Epoch 28/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 47973474304.0000 -
root_mean_squared_error: 217848.9062 - val_loss: 94834302976.0000 -
val_root_mean_squared_error: 308160.0938
Epoch 29/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 46558228480.0000 -
```

```
root_mean_squared_error: 214984.2344 - val_loss: 121336864768.0000 -
val_root_mean_squared_error: 348735.0625
Epoch 30/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 47827804160.0000 -
root_mean_squared_error: 217569.0156 - val_loss: 119992819712.0000 -
val_root_mean_squared_error: 346598.3750
Epoch 31/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42791772160.0000 -
root_mean_squared_error: 205805.7500 - val_loss: 111535497216.0000 -
val_root_mean_squared_error: 333928.3125
Epoch 32/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 80407166976.0000 -
root_mean_squared_error: 275982.7500 - val_loss: 94003757056.0000 -
val_root_mean_squared_error: 306557.1250
Epoch 33/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 32372895744.0000 -
root_mean_squared_error: 179877.3594 - val_loss: 85094948864.0000 -
val_root_mean_squared_error: 291727.2500
Epoch 34/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 23048769536.0000 -
root_mean_squared_error: 151735.9688 - val_loss: 94367162368.0000 -
val_root_mean_squared_error: 307271.3125
Epoch 35/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20433405952.0000 -
root_mean_squared_error: 142883.4688 - val_loss: 90934247424.0000 -
val_root_mean_squared_error: 301394.8125
Epoch 36/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 18574284800.0000 -
root_mean_squared_error: 136155.7969 - val_loss: 91156594688.0000 -
val_root_mean_squared_error: 302071.0625
Epoch 37/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 16524505088.0000 -
root_mean_squared_error: 128364.1875 - val_loss: 93396566016.0000 -
val_root_mean_squared_error: 305340.1250
Epoch 38/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 16133852160.0000 -
root_mean_squared_error: 126762.7891 - val_loss: 91973918720.0000 -
val_root_mean_squared_error: 303128.6875
Epoch 39/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 15483287552.0000 -
root_mean_squared_error: 124266.2500 - val_loss: 95709446144.0000 -
val_root_mean_squared_error: 309089.1562
Epoch 40/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 14578180096.0000 -
root_mean_squared_error: 120592.1953 - val_loss: 98561654784.0000 -
val_root_mean_squared_error: 313707.5625
Epoch 41/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 15841553408.0000 -
root_mean_squared_error: 125572.2969 - val_loss: 95438798848.0000 -
```

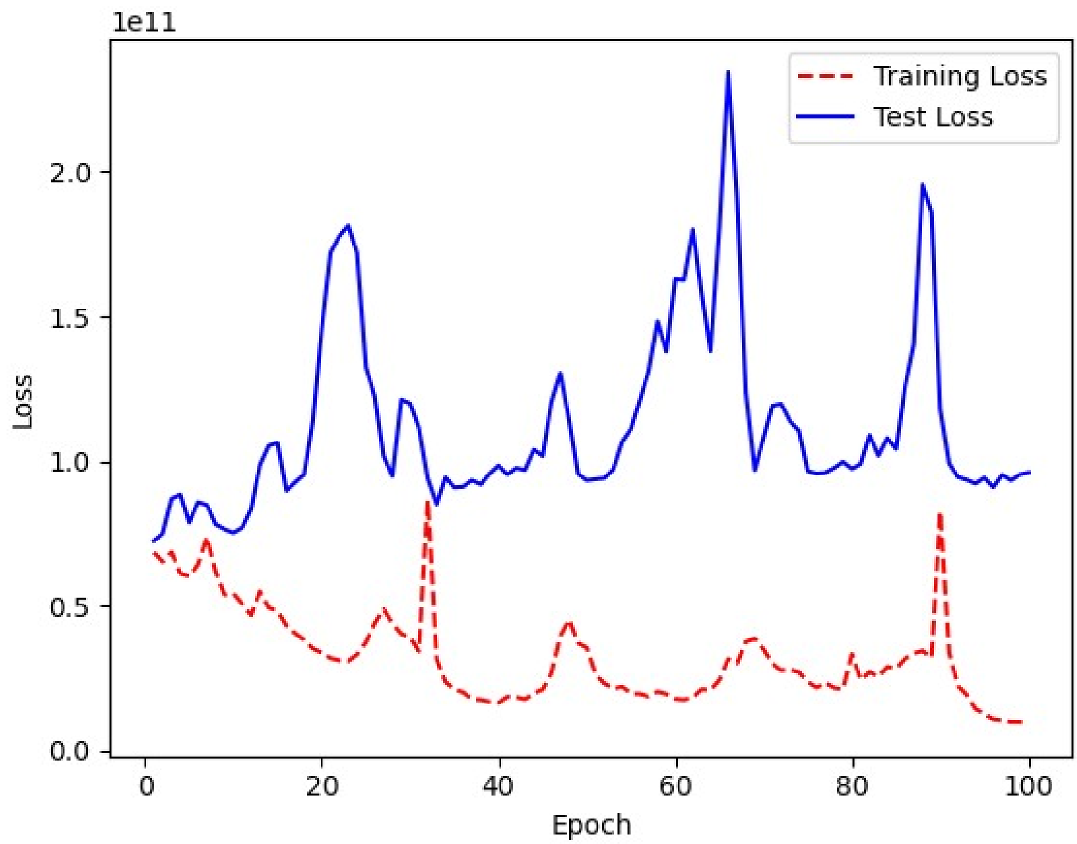
```
val_root_mean_squared_error: 308420.0938
Epoch 42/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 15873474560.0000 -
root_mean_squared_error: 125656.8359 - val_loss: 97700126720.0000 -
val_root_mean_squared_error: 312695.1875
Epoch 43/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 15610659840.0000 -
root_mean_squared_error: 124569.8828 - val_loss: 96966254592.0000 -
val_root_mean_squared_error: 311236.5625
Epoch 44/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 16079316992.0000 -
root_mean_squared_error: 126249.9844 - val_loss: 104021499904.0000 -
val_root_mean_squared_error: 322392.3125
Epoch 45/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 17827260416.0000 -
root_mean_squared_error: 132712.4688 - val_loss: 101800919040.0000 -
val_root_mean_squared_error: 319029.3125
Epoch 46/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20882669568.0000 -
root_mean_squared_error: 143205.0781 - val_loss: 120419500032.0000 -
val_root_mean_squared_error: 347312.8438
Epoch 47/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 30739800064.0000 -
root_mean_squared_error: 173888.9531 - val_loss: 130490998784.0000 -
val_root_mean_squared_error: 361702.5938
Epoch 48/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 42872123392.0000 -
root_mean_squared_error: 206784.1406 - val_loss: 114060263424.0000 -
val_root_mean_squared_error: 338098.6562
Epoch 49/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 38375317504.0000 -
root_mean_squared_error: 195701.6562 - val_loss: 95601483776.0000 -
val_root_mean_squared_error: 309518.6562
Epoch 50/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 42306641920.0000 -
root_mean_squared_error: 204829.3125 - val_loss: 93370638336.0000 -
val_root_mean_squared_error: 305640.6562
Epoch 51/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 27306375168.0000 -
root_mean_squared_error: 164934.3438 - val_loss: 93808918528.0000 -
val_root_mean_squared_error: 306423.0938
Epoch 52/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22754328576.0000 -
root_mean_squared_error: 150595.6719 - val_loss: 94127423488.0000 -
val_root_mean_squared_error: 306938.9688
Epoch 53/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20890413056.0000 -
root_mean_squared_error: 144292.1875 - val_loss: 97291083776.0000 -
val_root_mean_squared_error: 311891.8750
```

```
Epoch 54/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22408595456.0000 -
root_mean_squared_error: 149453.2344 - val_loss: 106768277504.0000 -
val_root_mean_squared_error: 326939.5312
Epoch 55/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 19599093760.0000 -
root_mean_squared_error: 139715.9531 - val_loss: 111245041664.0000 -
val_root_mean_squared_error: 333626.8125
Epoch 56/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 18367580160.0000 -
root_mean_squared_error: 135424.0000 - val_loss: 121050177536.0000 -
val_root_mean_squared_error: 347828.4688
Epoch 57/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 18238945280.0000 -
root_mean_squared_error: 134862.6094 - val_loss: 131129458688.0000 -
val_root_mean_squared_error: 361863.1875
Epoch 58/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20369037312.0000 -
root_mean_squared_error: 142608.6094 - val_loss: 148322336768.0000 -
val_root_mean_squared_error: 384279.5938
Epoch 59/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 19444164608.0000 -
root_mean_squared_error: 139288.0000 - val_loss: 137841344512.0000 -
val_root_mean_squared_error: 370579.6562
Epoch 60/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 17451395072.0000 -
root_mean_squared_error: 132008.3125 - val_loss: 162856386560.0000 -
val_root_mean_squared_error: 403177.3125
Epoch 61/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 17653540864.0000 -
root_mean_squared_error: 132725.7500 - val_loss: 162548957184.0000 -
val_root_mean_squared_error: 402708.1875
Epoch 62/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 18823653376.0000 -
root_mean_squared_error: 136867.7812 - val_loss: 180104101888.0000 -
val_root_mean_squared_error: 423943.4688
Epoch 63/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 24401852416.0000 -
root_mean_squared_error: 155662.4219 - val_loss: 157815701504.0000 -
val_root_mean_squared_error: 396651.2500
Epoch 64/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20369852416.0000 -
root_mean_squared_error: 142530.0625 - val_loss: 137976561664.0000 -
val_root_mean_squared_error: 371148.9375
Epoch 65/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22993967104.0000 -
root_mean_squared_error: 151489.5938 - val_loss: 180584136704.0000 -
val_root_mean_squared_error: 424157.6875
Epoch 66/100
```

```
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 27006674944.0000 -
root_mean_squared_error: 163826.7812 - val_loss: 234545561600.0000 -
val_root_mean_squared_error: 483315.5938
Epoch 67/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 31787866112.0000 -
root_mean_squared_error: 177627.2344 - val_loss: 190791352320.0000 -
val_root_mean_squared_error: 436376.6562
Epoch 68/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 42123214848.0000 -
root_mean_squared_error: 204105.0312 - val_loss: 123978997760.0000 -
val_root_mean_squared_error: 352136.6875
Epoch 69/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 51205672960.0000 -
root_mean_squared_error: 224604.6250 - val_loss: 96891092992.0000 -
val_root_mean_squared_error: 311464.4375
Epoch 70/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 45626081280.0000 -
root_mean_squared_error: 212370.1562 - val_loss: 108143779840.0000 -
val_root_mean_squared_error: 328684.8125
Epoch 71/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 35210838016.0000 -
root_mean_squared_error: 186438.5781 - val_loss: 119187193856.0000 -
val_root_mean_squared_error: 345078.5625
Epoch 72/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 29590577152.0000 -
root_mean_squared_error: 171031.4531 - val_loss: 119845355520.0000 -
val_root_mean_squared_error: 346098.3438
Epoch 73/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 27594123264.0000 -
root_mean_squared_error: 165440.5938 - val_loss: 113804836864.0000 -
val_root_mean_squared_error: 337146.9062
Epoch 74/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 23912464384.0000 -
root_mean_squared_error: 153876.4375 - val_loss: 110573101056.0000 -
val_root_mean_squared_error: 332147.6875
Epoch 75/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 23942627328.0000 -
root_mean_squared_error: 154439.6250 - val_loss: 96609951744.0000 -
val_root_mean_squared_error: 310669.8125
Epoch 76/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 21074219008.0000 -
root_mean_squared_error: 144584.0938 - val_loss: 95720562688.0000 -
val_root_mean_squared_error: 309226.1562
Epoch 77/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22246963200.0000 -
root_mean_squared_error: 148109.9375 - val_loss: 96003309568.0000 -
val_root_mean_squared_error: 310108.5625
Epoch 78/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 19527573504.0000 -
```

```
root_mean_squared_error: 139057.2500 - val_loss: 97744134144.0000 -
val_root_mean_squared_error: 312685.7500
Epoch 79/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 18740848640.0000 -
root_mean_squared_error: 136130.9688 - val_loss: 99798073344.0000 -
val_root_mean_squared_error: 315889.8125
Epoch 80/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 25471275008.0000 -
root_mean_squared_error: 156467.3281 - val_loss: 97396056064.0000 -
val_root_mean_squared_error: 312238.3125
Epoch 81/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 20355934208.0000 -
root_mean_squared_error: 141939.5156 - val_loss: 99116351488.0000 -
val_root_mean_squared_error: 315098.4688
Epoch 82/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22505570304.0000 -
root_mean_squared_error: 149309.4219 - val_loss: 109052796928.0000 -
val_root_mean_squared_error: 330194.6875
Epoch 83/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 21992380416.0000 -
root_mean_squared_error: 147933.7500 - val_loss: 101889622016.0000 -
val_root_mean_squared_error: 319528.2500
Epoch 84/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 24229808128.0000 -
root_mean_squared_error: 155263.1875 - val_loss: 108094996480.0000 -
val_root_mean_squared_error: 329054.0312
Epoch 85/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 23525470208.0000 -
root_mean_squared_error: 152940.8438 - val_loss: 104201256960.0000 -
val_root_mean_squared_error: 323173.9688
Epoch 86/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 28334966784.0000 -
root_mean_squared_error: 168003.7031 - val_loss: 125490495488.0000 -
val_root_mean_squared_error: 354407.5938
Epoch 87/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 27497064448.0000 -
root_mean_squared_error: 165417.8125 - val_loss: 140326780928.0000 -
val_root_mean_squared_error: 374575.2500
Epoch 88/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 27553630208.0000 -
root_mean_squared_error: 165551.8750 - val_loss: 195534422016.0000 -
val_root_mean_squared_error: 441762.4688
Epoch 89/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 28784975872.0000 -
root_mean_squared_error: 169313.7344 - val_loss: 186155188224.0000 -
val_root_mean_squared_error: 431174.4688
Epoch 90/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 76350316544.0000 -
root_mean_squared_error: 267227.3750 - val_loss: 117948645376.0000 -
```

```
val_root_mean_squared_error: 343607.2812
Epoch 91/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 36313653248.0000 -
root_mean_squared_error: 190397.5625 - val_loss: 99248136192.0000 -
val_root_mean_squared_error: 315244.1875
Epoch 92/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 23255791616.0000 -
root_mean_squared_error: 152318.0625 - val_loss: 94537228288.0000 -
val_root_mean_squared_error: 307671.5938
Epoch 93/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 18679957504.0000 -
root_mean_squared_error: 136516.8125 - val_loss: 93514661888.0000 -
val_root_mean_squared_error: 305891.8125
Epoch 94/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 13499862016.0000 -
root_mean_squared_error: 116071.0391 - val_loss: 92266807296.0000 -
val_root_mean_squared_error: 303799.6875
Epoch 95/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 11712325632.0000 -
root_mean_squared_error: 108041.0312 - val_loss: 94335664128.0000 -
val_root_mean_squared_error: 307213.5000
Epoch 96/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 10214775808.0000 -
root_mean_squared_error: 101014.3828 - val_loss: 91016552448.0000 -
val_root_mean_squared_error: 301714.1562
Epoch 97/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9393400832.0000 -
root_mean_squared_error: 96687.1328 - val_loss: 95347097600.0000 -
val_root_mean_squared_error: 308766.5000
Epoch 98/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 9029524480.0000 -
root_mean_squared_error: 94939.3125 - val_loss: 93341966336.0000 -
val_root_mean_squared_error: 305544.6875
Epoch 99/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8952472576.0000 -
root_mean_squared_error: 94378.9297 - val_loss: 95408046080.0000 -
val_root_mean_squared_error: 308887.5312
Epoch 100/100
219/219 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8684326912.0000 -
root_mean_squared_error: 93153.3672 - val_loss: 96090570752.0000 -
val_root_mean_squared_error: 309840.2812

# Visualize loss history for the large neural network
plt.plot(epoch_count, training_loss, 'r--')
plt.plot(epoch_count, test_loss, 'b-')
plt.legend(['Training Loss', 'Test Loss'])
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.show();
```




```
mse(gbr.predict(X_test), y_test, squared=False)
```

```
272790.7130232794
```

Methods to reduce overfitting refer to this link:

https://datascience.stackexchange.com/questions/65471/validation-loss-much-higher-than-training-loss

Huge credit for these legends that helped me with this project.

Author: DanB (Melbourne Housing Snapshot)

NeuralNine: https://www.youtube.com/watch?v=Wqmtf9SA_kk&t=676s

Greg Hogg: https://www.youtube.com/watch?v=_-UCcuB8nbw