Take the Lyons housing data set from last week and:

-build a price predictor in keras

-increase the model complexity until you have clear overfitting

-use L1, L2 and dropout to try to reduce overfitting, what method seems to work best for you?

-develop a "good" model and then use Shap to explain how it arrives at an answer.

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

In [6]:
```
#load the data, change the file address

infile= "lyon_housing.csv"
lyon=pd.read_csv(infile)
```

In [7]:
```
lyon.head()
```

Out[7]:

| | date_transaction | type_achat | type_bien | nombre_pieces | surface_logement | surface_carrez_logeme |
|---|---|---|---|---|---|---|
| **0** | 2019-10-31 | ancien | maison | 5 | 100.0 | Na |
| **1** | 2018-11-26 | ancien | maison | 2 | 52.0 | Na |
| **2** | 2016-08-04 | ancien | appartement | 1 | 28.0 | 28 |
| **3** | 2016-11-18 | ancien | appartement | 3 | 67.0 | 66 |
| **4** | 2016-12-16 | ancien | appartement | 1 | 28.0 | Na |

# So predict pridict prix with keraslink text

Do everything similar except use keras when required.

Build the model and is keras and put it aside to clean up the data.

No need to recode everything just copy and paste all code from "Lyons_Housing_Data.ipynb" but change things up where applicable for keras.

# Copying starts

In [8]:
```
lyon['date_transaction']=pd.to_datetime(lyon['date_transaction'])
```

In [9]:
```
lyon['date_transaction'].head()
```

Out[9]:
```
0   2019-10-31
1   2018-11-26
2   2016-08-04
3   2016-11-18
4   2016-12-16
Name: date_transaction, dtype: datetime64[ns]
```

In [10]:
```
lyon['year_transaction']=lyon['date_transaction'].dt.year
```

In [11]:
```
lyon['date_construction']=pd.to_datetime(lyon['date_construction'])
```

In [12]:
```
lyon['year_construction']=lyon['date_construction'].dt.year.astype(int)
```

In [13]:
```
lyon.head()
```

Out[13]:

| | date_transaction | type_achat | type_bien | nombre_pieces | surface_logement | surface_carrez_logeme |
|---|---|---|---|---|---|---|
| **0** | 2019-10-31 | ancien | maison | 5 | 100.0 | Na |
| **1** | 2018-11-26 | ancien | maison | 2 | 52.0 | Na |
| **2** | 2016-08-04 | ancien | appartement | 1 | 28.0 | 28 |
| **3** | 2016-11-18 | ancien | appartement | 3 | 67.0 | 66 |
| **4** | 2016-12-16 | ancien | appartement | 1 | 28.0 | Na |

In [14]:
```
lyon['anciennete'].describe()
```

Out[14]:
```
count    40516.000000
mean        21.246938
std          9.397379
min         -3.853563
25%         15.064690
50%         26.571388
75%         28.775403
max         31.494144
Name: anciennete, dtype: float64
```

In [15]:
```
temp=pd.cut(lyon.anciennete,bins=[-5,0,5,10,20,30,40],labels=['UnderConstruction','0-5
```

In [16]:
```
lyon['age']=temp
```

In [17]:
```
lyon.head(3)
```

Out[17]:

| | date_transaction | type_achat | type_bien | nombre_pieces | surface_logement | surface_carrez_logeme |
|---|---|---|---|---|---|---|
| **0** | 2019-10-31 | ancien | maison | 5 | 100.0 | Na |
| **1** | 2018-11-26 | ancien | maison | 2 | 52.0 | Na |
| **2** | 2016-08-04 | ancien | appartement | 1 | 28.0 | 28 |

In [18]:
```
lyon = lyon.drop(labels="surface_terrain", axis=1)
```

In [19]:
```
lyon.head(3)
```

Out[19]:

| | date_transaction | type_achat | type_bien | nombre_pieces | surface_logement | surface_carrez_logeme |
|---|---|---|---|---|---|---|
| **0** | 2019-10-31 | ancien | maison | 5 | 100.0 | Na |
| **1** | 2018-11-26 | ancien | maison | 2 | 52.0 | Na |
| **2** | 2016-08-04 | ancien | appartement | 1 | 28.0 | 28 |

In [20]:
```
lyon = lyon.drop(labels="surface_carrez_logement", axis=1)
```

In [20]:
```

```

In [21]:
```
#going to to impute all strings with the most frequent string
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer

#Impute catagorical data

# I should keep commune out perhaps as they are all commune, however I will keep it in
# There is also no nan values in that column but I just need it in the data frame

imp = SimpleImputer(strategy="most_frequent")
StringImpuing= pd.DataFrame(imp.fit_transform(lyon[["type_achat","type_bien","commune"

#Impute Continuous Data with the median value

impCont = SimpleImputer(strategy="median")
continuousImputing= pd.DataFrame(impCont.fit_transform(lyon[["nombre_pieces","surface_
```

In [22]:
```
StringImpuing.head()
```

Out[22]:

| | type_achat | type_bien | commune | age |
|---|---|---|---|---|
| **0** | ancien | maison | Villeurbanne | 10-20 |
| **1** | ancien | maison | Villeurbanne | 10-20 |
| **2** | ancien | appartement | Villeurbanne | 10-20 |
| **3** | ancien | appartement | Villeurbanne | 10-20 |
| **4** | ancien | appartement | Villeurbanne | 10-20 |

In [23]:
```
continuousImputing.head()
```

Out[23]:

| | nombre_pieces | surface_logement | nombre_parkings |
|---|---|---|---|
| **0** | 5.0 | 100.0 | 0.0 |
| **1** | 2.0 | 52.0 | 0.0 |
| **2** | 1.0 | 28.0 | 1.0 |
| **3** | 3.0 | 67.0 | 1.0 |
| **4** | 1.0 | 28.0 | 1.0 |

In [24]:
```
# Combine string imouting with year transaaction

CatagoricalVariable = pd.concat([StringImpuing,lyon["year_transaction"]],axis = 1)
CatagoricalVariable["year_transaction"]
```

Out[24]:
```
0        2019
1        2018
2        2016
3        2016
4        2016
         ...
40511    2020
40512    2020
40513    2020
40514    2020
40515    2020
Name: year_transaction, Length: 40516, dtype: int64
```

In [25]:
```
StringImpuing.head()
```

Out[25]:

| | type_achat | type_bien | commune | age |
|---|---|---|---|---|
| **0** | ancien | maison | Villeurbanne | 10-20 |
| **1** | ancien | maison | Villeurbanne | 10-20 |
| **2** | ancien | appartement | Villeurbanne | 10-20 |
| **3** | ancien | appartement | Villeurbanne | 10-20 |
| **4** | ancien | appartement | Villeurbanne | 10-20 |

In [26]:
```
CatagoricalVariable.head()
```

Out[26]:

| | type_achat | type_bien | commune | age | year_transaction |
|---|---|---|---|---|---|
| **0** | ancien | maison | Villeurbanne | 10-20 | 2019 |
| **1** | ancien | maison | Villeurbanne | 10-20 | 2018 |
| **2** | ancien | appartement | Villeurbanne | 10-20 | 2016 |
| **3** | ancien | appartement | Villeurbanne | 10-20 | 2016 |
| **4** | ancien | appartement | Villeurbanne | 10-20 | 2016 |

# one hot encode the categories

In [27]:
```
from sklearn.preprocessing import OneHotEncoder

encode_lyon=OneHotEncoder()

encode_lyon_fit1= encode_lyon.fit_transform(lyon[CatagoricalVariable.columns[0]].to_nu
df_type_achat=pd.DataFrame(encode_lyon_fit1.toarray(),columns=encode_lyon.categories_[

encode_lyon_fit2= encode_lyon.fit_transform(lyon[CatagoricalVariable.columns[1]].to_nu
df_type_2=pd.DataFrame(encode_lyon_fit2.toarray(),columns=encode_lyon.categories_[0][:

encode_lyon_fit3= encode_lyon.fit_transform(lyon[CatagoricalVariable.columns[2]].to_nu
df_type_3=pd.DataFrame(encode_lyon_fit3.toarray(),columns=encode_lyon.categories_[0][:

encode_lyon_fit4= encode_lyon.fit_transform(lyon[CatagoricalVariable.columns[3]].to_nu
```

```
df_type_4=pd.DataFrame(encode_lyon_fit4.toarray(),columns=encode_lyon.categories_[0][:



encode_lyon_fit5= encode_lyon.fit_transform(lyon[CatagoricalVariable.columns[4]].to_nu
stringCon= encode_lyon.categories_[0][:].astype(str)
df_type_5=pd.DataFrame(encode_lyon_fit5.toarray(),columns=encode_lyon.categories_[0][:

df_type_5.columns=stringCon
```

In [27]:

In [28]:
```
# concatenate these one-hot-encoded versions of the categorical variables

df_cats_lyon=pd.concat([df_type_achat,df_type_2,df_type_3,df_type_4,df_type_5], axis=1
df_cats_lyon.head()
```

Out[28]:

| | VEFA | ancien | appartement | maison | Lyon 1er Arrondissement | Lyon 2e Arrondissement | Lyon 3e Arrondissement | Lyc Arrondisse |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| **1** | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| **2** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **3** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **4** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

5 rows × 26 columns

# Standardization

# standard scale the other data

In [29]:
```
scaler = StandardScaler()

lyons_continuous = scaler.fit_transform(continuousImputing)
lyons_continuous=pd.DataFrame(lyons_continuous,columns=continuousImputing.columns)
lyons_continuous
```

Out[29]:

| | nombre_pieces | surface_logement | nombre_parkings |
|---|---|---|---|
| **0** | 1.870396 | 1.231567 | -0.996559 |
| **1** | -0.671310 | -0.469113 | -0.996559 |
| **2** | -1.518546 | -1.319453 | 0.666260 |
| **3** | 0.175925 | 0.062349 | 0.666260 |
| **4** | -1.518546 | -1.319453 | 0.666260 |
| **...** | ... | ... | ... |
| **40511** | -0.671310 | -1.106868 | 0.666260 |
| **40512** | -0.671310 | -1.142299 | -0.996559 |
| **40513** | -0.671310 | -1.496607 | -0.996559 |
| **40514** | -0.671310 | -1.106868 | -0.996559 |
| **40515** | -0.671310 | -1.248592 | -0.996559 |

40516 rows × 3 columns

In [30]:
```
lyon_housing_final = pd.concat([df_cats_lyon,lyons_continuous],axis=1)
```

In [31]:
```
lyon_housing_final
```

Out[31]:

| | VEFA | ancien | appartement | maison | Lyon 1er Arrondissement | Lyon 2e Arrondissement | Lyon 3e Arrondissement | Arron |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| **1** | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| **2** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **3** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **4** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **40511** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **40512** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **40513** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **40514** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **40515** | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

40516 rows × 29 columns

In [32]:
```
y=lyon.loc[:,'prix']
```

In [33]:
```
y.head()
```

Out[33]:
```
0    530000.0
1    328550.0
2     42500.0
3    180900.0
4     97000.0
Name: prix, dtype: float64
```

In [34]:
```
lyon_housing_final = lyon_housing_final.to_numpy()
```

-build a price predictor in keras

In [35]:
```
# split off a validation set from the training data
from sklearn.model_selection import train_test_split

X_train2, X_valid, y_train2, y_valid = train_test_split(lyon_housing_final, y, test_si
```

In [36]:
```
lyon_housing_final[0]
```

Out[36]:
```
array([ 0.        ,  1.        ,  0.        ,  1.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  1.        ,  0.        ,
        1.        ,  0.        ,  0.        ,  0.        ,  0.        ,
        0.        ,  0.        ,  0.        ,  1.        ,  0.        ,
        0.        ,  1.87039624,  1.23156693, -0.9965586 ])
```

In [37]:
```
y
```

Out[37]:
```
0        530000.0
1        328550.0
2         42500.0
3        180900.0
4         97000.0
           ...
40511    226000.0
40512    217300.0
40513    145000.0
40514    206000.0
40515    187500.0
Name: prix, Length: 40516, dtype: float64
```

# Build the model and fit the model

In [45]:
```
from tensorflow.keras import models
from tensorflow.keras import layers

def build_model():
    model=models.Sequential()
    model.add(layers.Dense(64,activation="relu",input_shape=(lyon_housing_final.shape[
    model.add(layers.Dense(64,activation="relu"))
    model.add(layers.Dense(64,activation='relu'))
    model.add(layers.Dense(10,activation='softmax'))
    model.add(layers.Dense(1))

    model.compile(optimizer='rmsprop',loss='mse',metrics=['mse','mae'])
    return(model)
```

In [46]:
```
from tensorflow.keras.callbacks import History
history = History()
```

In [48]:
```
model = build_model()
# Train the model (in silent mode, verbose=0)
num_epochs=100

history= model.fit(X_train2, y_train2,epochs=num_epochs, batch_size=64, verbose=1, val
```

```
Epoch 1/100
507/507 [==============================] - 4s 5ms/step - loss: 89136414720.0000 - ms
e: 89136414720.0000 - mae: 255683.9531 - val_loss: 89852305408.0000 - val_mse: 898523
05408.0000 - val_mae: 256052.5156
Epoch 2/100
507/507 [==============================] - 2s 4ms/step - loss: 89135857664.0000 - ms
e: 89135857664.0000 - mae: 255682.8594 - val_loss: 89851781120.0000 - val_mse: 898517
81120.0000 - val_mae: 256051.4531
Epoch 3/100
507/507 [==============================] - 2s 4ms/step - loss: 89135308800.0000 - ms
e: 89135308800.0000 - mae: 255681.9062 - val_loss: 89851240448.0000 - val_mse: 898512
40448.0000 - val_mae: 256050.5000
Epoch 4/100
507/507 [==============================] - 3s 5ms/step - loss: 89134891008.0000 - ms
e: 89134891008.0000 - mae: 255680.9062 - val_loss: 89850732544.0000 - val_mse: 898507
32544.0000 - val_mae: 256049.4531
Epoch 5/100
507/507 [==============================] - 2s 5ms/step - loss: 89134309376.0000 - ms
e: 89134309376.0000 - mae: 255679.9219 - val_loss: 89850232832.0000 - val_mse: 898502
32832.0000 - val_mae: 256048.4375
Epoch 6/100
507/507 [==============================] - 2s 4ms/step - loss: 89133760512.0000 - ms
e: 89133760512.0000 - mae: 255679.0312 - val_loss: 89849716736.0000 - val_mse: 898497
16736.0000 - val_mae: 256047.4219
Epoch 7/100
507/507 [==============================] - 2s 4ms/step - loss: 89133268992.0000 - ms
e: 89133268992.0000 - mae: 255677.9375 - val_loss: 89849208832.0000 - val_mse: 898492
08832.0000 - val_mae: 256046.4531
Epoch 8/100
507/507 [==============================] - 2s 4ms/step - loss: 89132793856.0000 - ms
e: 89132793856.0000 - mae: 255677.0312 - val_loss: 89848659968.0000 - val_mse: 898486
59968.0000 - val_mae: 256045.4062
Epoch 9/100
507/507 [==============================] - 2s 4ms/step - loss: 89132220416.0000 - ms
e: 89132220416.0000 - mae: 255675.8594 - val_loss: 89848152064.0000 - val_mse: 898481
52064.0000 - val_mae: 256044.4531
Epoch 10/100
507/507 [==============================] - 3s 6ms/step - loss: 89131778048.0000 - ms
e: 89131778048.0000 - mae: 255674.9375 - val_loss: 89847619584.0000 - val_mse: 898476
19584.0000 - val_mae: 256043.4062
Epoch 11/100
507/507 [==============================] - 2s 4ms/step - loss: 89131245568.0000 - ms
e: 89131245568.0000 - mae: 255673.9219 - val_loss: 89847119872.0000 - val_mse: 898471
19872.0000 - val_mae: 256042.4062
Epoch 12/100
507/507 [==============================] - 2s 4ms/step - loss: 89130680320.0000 - ms
e: 89130680320.0000 - mae: 255672.8438 - val_loss: 89846579200.0000 - val_mse: 898465
79200.0000 - val_mae: 256041.3750
Epoch 13/100
507/507 [==============================] - 2s 4ms/step - loss: 89130205184.0000 - ms
e: 89130205184.0000 - mae: 255671.7969 - val_loss: 89846087680.0000 - val_mse: 898460
87680.0000 - val_mae: 256040.4219
Epoch 14/100
507/507 [==============================] - 2s 4ms/step - loss: 89129697280.0000 - ms
e: 89129697280.0000 - mae: 255670.7188 - val_loss: 89845555200.0000 - val_mse: 898455
55200.0000 - val_mae: 256039.3594
Epoch 15/100
507/507 [==============================] - 3s 6ms/step - loss: 89129164800.0000 - ms
e: 89129164800.0000 - mae: 255669.8750 - val_loss: 89845047296.0000 - val_mse: 898450
47296.0000 - val_mae: 256038.3750
```

```
Epoch 16/100
507/507 [==============================] - 2s 4ms/step - loss: 89128599552.0000 - ms
e: 89128599552.0000 - mae: 255668.7344 - val_loss: 89844531200.0000 - val_mse: 898445
31200.0000 - val_mae: 256037.3594
Epoch 17/100
507/507 [==============================] - 2s 4ms/step - loss: 89128108032.0000 - ms
e: 89128108032.0000 - mae: 255667.8438 - val_loss: 89844006912.0000 - val_mse: 898440
06912.0000 - val_mae: 256036.3750
Epoch 18/100
507/507 [==============================] - 2s 4ms/step - loss: 89127567360.0000 - ms
e: 89127567360.0000 - mae: 255666.8281 - val_loss: 89843466240.0000 - val_mse: 898434
66240.0000 - val_mae: 256035.3125
Epoch 19/100
507/507 [==============================] - 3s 6ms/step - loss: 89127100416.0000 - ms
e: 89127100416.0000 - mae: 255665.7969 - val_loss: 89842974720.0000 - val_mse: 898429
74720.0000 - val_mae: 256034.3594
Epoch 20/100
507/507 [==============================] - 3s 6ms/step - loss: 89126576128.0000 - ms
e: 89126576128.0000 - mae: 255664.7500 - val_loss: 89842434048.0000 - val_mse: 898424
34048.0000 - val_mae: 256033.2969
Epoch 21/100
507/507 [==============================] - 2s 4ms/step - loss: 89126019072.0000 - ms
e: 89126019072.0000 - mae: 255663.7500 - val_loss: 89841950720.0000 - val_mse: 898419
50720.0000 - val_mae: 256032.2969
Epoch 22/100
507/507 [==============================] - 2s 4ms/step - loss: 89125552128.0000 - ms
e: 89125552128.0000 - mae: 255662.8125 - val_loss: 89841410048.0000 - val_mse: 898414
10048.0000 - val_mae: 256031.2188
Epoch 23/100
507/507 [==============================] - 2s 4ms/step - loss: 89125076992.0000 - ms
e: 89125076992.0000 - mae: 255661.7500 - val_loss: 89840918528.0000 - val_mse: 898409
18528.0000 - val_mae: 256030.2812
Epoch 24/100
507/507 [==============================] - 2s 4ms/step - loss: 89124503552.0000 - ms
e: 89124503552.0000 - mae: 255660.6562 - val_loss: 89840369664.0000 - val_mse: 898403
69664.0000 - val_mae: 256029.2344
Epoch 25/100
507/507 [==============================] - 2s 5ms/step - loss: 89123979264.0000 - ms
e: 89123979264.0000 - mae: 255659.6562 - val_loss: 89839861760.0000 - val_mse: 898398
61760.0000 - val_mae: 256028.2344
Epoch 26/100
507/507 [==============================] - 3s 5ms/step - loss: 89123528704.0000 - ms
e: 89123528704.0000 - mae: 255658.7969 - val_loss: 89839345664.0000 - val_mse: 898393
45664.0000 - val_mae: 256027.2188
Epoch 27/100
507/507 [==============================] - 2s 4ms/step - loss: 89122947072.0000 - ms
e: 89122947072.0000 - mae: 255657.6562 - val_loss: 89838829568.0000 - val_mse: 898388
29568.0000 - val_mae: 256026.1875
Epoch 28/100
507/507 [==============================] - 2s 4ms/step - loss: 89122439168.0000 - ms
e: 89122439168.0000 - mae: 255656.5469 - val_loss: 89838313472.0000 - val_mse: 898383
13472.0000 - val_mae: 256025.1562
Epoch 29/100
507/507 [==============================] - 3s 6ms/step - loss: 89121931264.0000 - ms
e: 89121931264.0000 - mae: 255655.6875 - val_loss: 89837805568.0000 - val_mse: 898378
05568.0000 - val_mae: 256024.2031
Epoch 30/100
507/507 [==============================] - 5s 10ms/step - loss: 89121415168.0000 - ms
e: 89121415168.0000 - mae: 255654.6562 - val_loss: 89837281280.0000 - val_mse: 898372
81280.0000 - val_mae: 256023.1562
```

```
Epoch 31/100
507/507 [==============================] - 4s 8ms/step - loss: 89120841728.0000 - ms
e: 89120841728.0000 - mae: 255653.5469 - val_loss: 89836756992.0000 - val_mse: 898367
56992.0000 - val_mae: 256022.1562
Epoch 32/100
507/507 [==============================] - 3s 7ms/step - loss: 89120358400.0000 - ms
e: 89120358400.0000 - mae: 255652.7969 - val_loss: 89836240896.0000 - val_mse: 898362
40896.0000 - val_mae: 256021.1250
Epoch 33/100
507/507 [==============================] - 2s 4ms/step - loss: 89119809536.0000 - ms
e: 89119809536.0000 - mae: 255651.6094 - val_loss: 89835708416.0000 - val_mse: 898357
08416.0000 - val_mae: 256020.1406
Epoch 34/100
507/507 [==============================] - 3s 5ms/step - loss: 89119301632.0000 - ms
e: 89119301632.0000 - mae: 255650.6562 - val_loss: 89835216896.0000 - val_mse: 898352
16896.0000 - val_mae: 256019.1250
Epoch 35/100
507/507 [==============================] - 2s 4ms/step - loss: 89118851072.0000 - ms
e: 89118851072.0000 - mae: 255649.5312 - val_loss: 89834692608.0000 - val_mse: 898346
92608.0000 - val_mae: 256018.1094
Epoch 36/100
507/507 [==============================] - 2s 4ms/step - loss: 89118326784.0000 - ms
e: 89118326784.0000 - mae: 255648.5312 - val_loss: 89834160128.0000 - val_mse: 898341
60128.0000 - val_mae: 256017.0781
Epoch 37/100
507/507 [==============================] - 2s 4ms/step - loss: 89117827072.0000 - ms
e: 89117827072.0000 - mae: 255647.5781 - val_loss: 89833644032.0000 - val_mse: 898336
44032.0000 - val_mae: 256016.1094
Epoch 38/100
507/507 [==============================] - 2s 4ms/step - loss: 89117319168.0000 - ms
e: 89117319168.0000 - mae: 255646.6562 - val_loss: 89833119744.0000 - val_mse: 898331
19744.0000 - val_mae: 256015.0469
Epoch 39/100
507/507 [==============================] - 3s 6ms/step - loss: 89116737536.0000 - ms
e: 89116737536.0000 - mae: 255645.4219 - val_loss: 89832611840.0000 - val_mse: 898326
11840.0000 - val_mae: 256014.0625
Epoch 40/100
507/507 [==============================] - 2s 4ms/step - loss: 89116262400.0000 - ms
e: 89116262400.0000 - mae: 255644.4688 - val_loss: 89832087552.0000 - val_mse: 898320
87552.0000 - val_mae: 256013.0156
Epoch 41/100
507/507 [==============================] - 2s 4ms/step - loss: 89115738112.0000 - ms
e: 89115738112.0000 - mae: 255643.5938 - val_loss: 89831579648.0000 - val_mse: 898315
79648.0000 - val_mae: 256012.0469
Epoch 42/100
507/507 [==============================] - 2s 4ms/step - loss: 89115181056.0000 - ms
e: 89115181056.0000 - mae: 255642.2969 - val_loss: 89831055360.0000 - val_mse: 898310
55360.0000 - val_mae: 256010.9844
Epoch 43/100
507/507 [==============================] - 2s 4ms/step - loss: 89114697728.0000 - ms
e: 89114697728.0000 - mae: 255641.5625 - val_loss: 89830514688.0000 - val_mse: 898305
14688.0000 - val_mae: 256009.9844
Epoch 44/100
507/507 [==============================] - 3s 5ms/step - loss: 89114181632.0000 - ms
e: 89114181632.0000 - mae: 255640.7656 - val_loss: 89830039552.0000 - val_mse: 898300
39552.0000 - val_mae: 256008.9531
Epoch 45/100
507/507 [==============================] - 2s 5ms/step - loss: 89113632768.0000 - ms
e: 89113632768.0000 - mae: 255639.4844 - val_loss: 89829490688.0000 - val_mse: 898294
90688.0000 - val_mae: 256007.9375
```

```
Epoch 46/100
507/507 [==============================] - 2s 4ms/step - loss: 89113124864.0000 - ms
e: 89113124864.0000 - mae: 255638.4531 - val_loss: 89828999168.0000 - val_mse: 898289
99168.0000 - val_mae: 256006.9375
Epoch 47/100
507/507 [==============================] - 2s 4ms/step - loss: 89112608768.0000 - ms
e: 89112608768.0000 - mae: 255637.3750 - val_loss: 89828466688.0000 - val_mse: 898284
66688.0000 - val_mae: 256005.9531
Epoch 48/100
507/507 [==============================] - 2s 4ms/step - loss: 89112125440.0000 - ms
e: 89112125440.0000 - mae: 255636.3438 - val_loss: 89827958784.0000 - val_mse: 898279
58784.0000 - val_mae: 256004.9062
Epoch 49/100
507/507 [==============================] - 3s 5ms/step - loss: 89111568384.0000 - ms
e: 89111568384.0000 - mae: 255635.5000 - val_loss: 89827450880.0000 - val_mse: 898274
50880.0000 - val_mae: 256003.9219
Epoch 50/100
507/507 [==============================] - 3s 5ms/step - loss: 89111060480.0000 - ms
e: 89111060480.0000 - mae: 255634.3281 - val_loss: 89826918400.0000 - val_mse: 898269
18400.0000 - val_mae: 256002.9062
Epoch 51/100
507/507 [==============================] - 2s 4ms/step - loss: 89110560768.0000 - ms
e: 89110560768.0000 - mae: 255633.4062 - val_loss: 89826426880.0000 - val_mse: 898264
26880.0000 - val_mae: 256001.8906
Epoch 52/100
507/507 [==============================] - 2s 4ms/step - loss: 89110036480.0000 - ms
e: 89110036480.0000 - mae: 255632.4062 - val_loss: 89825869824.0000 - val_mse: 898258
69824.0000 - val_mae: 256000.8906
Epoch 53/100
507/507 [==============================] - 2s 4ms/step - loss: 89109544960.0000 - ms
e: 89109544960.0000 - mae: 255631.3750 - val_loss: 89825394688.0000 - val_mse: 898253
94688.0000 - val_mae: 255999.8906
Epoch 54/100
507/507 [==============================] - 2s 4ms/step - loss: 89108955136.0000 - ms
e: 89108955136.0000 - mae: 255630.4531 - val_loss: 89824821248.0000 - val_mse: 898248
21248.0000 - val_mae: 255998.8906
Epoch 55/100
507/507 [==============================] - 3s 6ms/step - loss: 89108537344.0000 - ms
e: 89108537344.0000 - mae: 255629.4375 - val_loss: 89824354304.0000 - val_mse: 898243
54304.0000 - val_mae: 255997.8906
Epoch 56/100
507/507 [==============================] - 2s 4ms/step - loss: 89107980288.0000 - ms
e: 89107980288.0000 - mae: 255628.2812 - val_loss: 89823813632.0000 - val_mse: 898238
13632.0000 - val_mae: 255996.8594
Epoch 57/100
507/507 [==============================] - 2s 4ms/step - loss: 89107415040.0000 - ms
e: 89107415040.0000 - mae: 255627.4531 - val_loss: 89823297536.0000 - val_mse: 898232
97536.0000 - val_mae: 255995.8594
Epoch 58/100
507/507 [==============================] - 2s 4ms/step - loss: 89106956288.0000 - ms
e: 89106956288.0000 - mae: 255626.3125 - val_loss: 89822781440.0000 - val_mse: 898227
81440.0000 - val_mae: 255994.8438
Epoch 59/100
507/507 [==============================] - 2s 4ms/step - loss: 89106407424.0000 - ms
e: 89106407424.0000 - mae: 255625.1719 - val_loss: 89822265344.0000 - val_mse: 898222
65344.0000 - val_mae: 255993.8438
Epoch 60/100
507/507 [==============================] - 3s 5ms/step - loss: 89105899520.0000 - ms
e: 89105899520.0000 - mae: 255624.3438 - val_loss: 89821757440.0000 - val_mse: 898217
57440.0000 - val_mae: 255992.8281
```

```
Epoch 61/100
507/507 [==============================] - 2s 5ms/step - loss: 89105375232.0000 - ms
e: 89105375232.0000 - mae: 255623.2812 - val_loss: 89821224960.0000 - val_mse: 898212
24960.0000 - val_mae: 255991.7969
Epoch 62/100
507/507 [==============================] - 2s 4ms/step - loss: 89104809984.0000 - ms
e: 89104809984.0000 - mae: 255622.3594 - val_loss: 89820717056.0000 - val_mse: 898207
17056.0000 - val_mae: 255990.7969
Epoch 63/100
507/507 [==============================] - 2s 4ms/step - loss: 89104293888.0000 - ms
e: 89104293888.0000 - mae: 255621.2969 - val_loss: 89820184576.0000 - val_mse: 898201
84576.0000 - val_mae: 255989.7812
Epoch 64/100
507/507 [==============================] - 2s 4ms/step - loss: 89103859712.0000 - ms
e: 89103859712.0000 - mae: 255620.3438 - val_loss: 89819676672.0000 - val_mse: 898196
76672.0000 - val_mae: 255988.7969
Epoch 65/100
507/507 [==============================] - 3s 5ms/step - loss: 89103278080.0000 - ms
e: 89103278080.0000 - mae: 255619.3906 - val_loss: 89819160576.0000 - val_mse: 898191
60576.0000 - val_mae: 255987.7656
Epoch 66/100
507/507 [==============================] - 3s 5ms/step - loss: 89102835712.0000 - ms
e: 89102835712.0000 - mae: 255618.3438 - val_loss: 89818660864.0000 - val_mse: 898186
60864.0000 - val_mae: 255986.7656
Epoch 67/100
507/507 [==============================] - 2s 4ms/step - loss: 89102295040.0000 - ms
e: 89102295040.0000 - mae: 255617.2031 - val_loss: 89818112000.0000 - val_mse: 898181
12000.0000 - val_mae: 255985.7188
Epoch 68/100
507/507 [==============================] - 2s 4ms/step - loss: 89101787136.0000 - ms
e: 89101787136.0000 - mae: 255616.2188 - val_loss: 89817604096.0000 - val_mse: 898176
04096.0000 - val_mae: 255984.7500
Epoch 69/100
507/507 [==============================] - 2s 4ms/step - loss: 89101221888.0000 - ms
e: 89101221888.0000 - mae: 255615.1562 - val_loss: 89817063424.0000 - val_mse: 898170
63424.0000 - val_mae: 255983.7188
Epoch 70/100
507/507 [==============================] - 2s 4ms/step - loss: 89100746752.0000 - ms
e: 89100746752.0000 - mae: 255614.2812 - val_loss: 89816588288.0000 - val_mse: 898165
88288.0000 - val_mae: 255982.7500
Epoch 71/100
507/507 [==============================] - 3s 6ms/step - loss: 89100247040.0000 - ms
e: 89100247040.0000 - mae: 255613.1719 - val_loss: 89816047616.0000 - val_mse: 898160
47616.0000 - val_mae: 255981.6719
Epoch 72/100
507/507 [==============================] - 2s 4ms/step - loss: 89099665408.0000 - ms
e: 89099665408.0000 - mae: 255612.2188 - val_loss: 89815564288.0000 - val_mse: 898155
64288.0000 - val_mae: 255980.7344
Epoch 73/100
507/507 [==============================] - 3s 6ms/step - loss: 89099141120.0000 - ms
e: 89099141120.0000 - mae: 255611.1250 - val_loss: 89815015424.0000 - val_mse: 898150
15424.0000 - val_mae: 255979.6719
Epoch 74/100
507/507 [==============================] - 2s 4ms/step - loss: 89098682368.0000 - ms
e: 89098682368.0000 - mae: 255610.2031 - val_loss: 89814523904.0000 - val_mse: 898145
23904.0000 - val_mae: 255978.7188
Epoch 75/100
507/507 [==============================] - 2s 4ms/step - loss: 89098174464.0000 - ms
e: 89098174464.0000 - mae: 255609.1094 - val_loss: 89813958656.0000 - val_mse: 898139
58656.0000 - val_mae: 255977.6719
```

```
Epoch 76/100
507/507 [==============================] - 3s 6ms/step - loss: 89097592832.0000 - ms
e: 89097592832.0000 - mae: 255608.2344 - val_loss: 89813483520.0000 - val_mse: 898134
83520.0000 - val_mae: 255976.7031
Epoch 77/100
507/507 [==============================] - 2s 4ms/step - loss: 89097068544.0000 - ms
e: 89097068544.0000 - mae: 255607.2344 - val_loss: 89812926464.0000 - val_mse: 898129
26464.0000 - val_mae: 255975.6406
Epoch 78/100
507/507 [==============================] - 2s 5ms/step - loss: 89096609792.0000 - ms
e: 89096609792.0000 - mae: 255606.1719 - val_loss: 89812459520.0000 - val_mse: 898124
59520.0000 - val_mae: 255974.6875
Epoch 79/100
507/507 [==============================] - 2s 4ms/step - loss: 89096077312.0000 - ms
e: 89096077312.0000 - mae: 255605.1719 - val_loss: 89811877888.0000 - val_mse: 898118
77888.0000 - val_mae: 255973.6094
Epoch 80/100
507/507 [==============================] - 2s 4ms/step - loss: 89095520256.0000 - ms
e: 89095520256.0000 - mae: 255604.3125 - val_loss: 89811419136.0000 - val_mse: 898114
19136.0000 - val_mae: 255972.6406
Epoch 81/100
507/507 [==============================] - 3s 5ms/step - loss: 89095004160.0000 - ms
e: 89095004160.0000 - mae: 255603.0625 - val_loss: 89810853888.0000 - val_mse: 898108
53888.0000 - val_mae: 255971.5938
Epoch 82/100
507/507 [==============================] - 3s 5ms/step - loss: 89094488064.0000 - ms
e: 89094488064.0000 - mae: 255602.1562 - val_loss: 89810370560.0000 - val_mse: 898103
70560.0000 - val_mae: 255970.6406
Epoch 83/100
507/507 [==============================] - 2s 4ms/step - loss: 89094004736.0000 - ms
e: 89094004736.0000 - mae: 255601.0312 - val_loss: 89809813504.0000 - val_mse: 898098
13504.0000 - val_mae: 255969.5938
Epoch 84/100
507/507 [==============================] - 2s 4ms/step - loss: 89093521408.0000 - ms
e: 89093521408.0000 - mae: 255599.9844 - val_loss: 89809330176.0000 - val_mse: 898093
30176.0000 - val_mae: 255968.6250
Epoch 85/100
507/507 [==============================] - 2s 4ms/step - loss: 89092964352.0000 - ms
e: 89092964352.0000 - mae: 255599.0625 - val_loss: 89808764928.0000 - val_mse: 898087
64928.0000 - val_mae: 255967.5312
Epoch 86/100
507/507 [==============================] - 2s 4ms/step - loss: 89092431872.0000 - ms
e: 89092431872.0000 - mae: 255598.1094 - val_loss: 89808289792.0000 - val_mse: 898082
89792.0000 - val_mae: 255966.6094
Epoch 87/100
507/507 [==============================] - 3s 5ms/step - loss: 89091891200.0000 - ms
e: 89091891200.0000 - mae: 255597.2656 - val_loss: 89807749120.0000 - val_mse: 898077
49120.0000 - val_mae: 255965.5156
Epoch 88/100
507/507 [==============================] - 2s 4ms/step - loss: 89091407872.0000 - ms
e: 89091407872.0000 - mae: 255596.0781 - val_loss: 89807265792.0000 - val_mse: 898072
65792.0000 - val_mae: 255964.5625
Epoch 89/100
507/507 [==============================] - 2s 4ms/step - loss: 89090867200.0000 - ms
e: 89090867200.0000 - mae: 255594.9688 - val_loss: 89806716928.0000 - val_mse: 898067
16928.0000 - val_mae: 255963.5000
Epoch 90/100
507/507 [==============================] - 2s 4ms/step - loss: 89090392064.0000 - ms
e: 89090392064.0000 - mae: 255593.8594 - val_loss: 89806225408.0000 - val_mse: 898062
25408.0000 - val_mae: 255962.5000
```

```
Epoch 91/100
507/507 [==============================] - 2s 4ms/step - loss: 89089859584.0000 - ms
e: 89089859584.0000 - mae: 255592.8750 - val_loss: 89805676544.0000 - val_mse: 898056
76544.0000 - val_mae: 255961.4688
Epoch 92/100
507/507 [==============================] - 3s 6ms/step - loss: 89089343488.0000 - ms
e: 89089343488.0000 - mae: 255591.9219 - val_loss: 89805201408.0000 - val_mse: 898052
01408.0000 - val_mae: 255960.5156
Epoch 93/100
507/507 [==============================] - 2s 4ms/step - loss: 89088827392.0000 - ms
e: 89088827392.0000 - mae: 255590.9844 - val_loss: 89804644352.0000 - val_mse: 898046
44352.0000 - val_mae: 255959.4688
Epoch 94/100
507/507 [==============================] - 2s 4ms/step - loss: 89088303104.0000 - ms
e: 89088303104.0000 - mae: 255589.9062 - val_loss: 89804169216.0000 - val_mse: 898041
69216.0000 - val_mae: 255958.4844
Epoch 95/100
507/507 [==============================] - 2s 4ms/step - loss: 89087803392.0000 - ms
e: 89087803392.0000 - mae: 255588.9219 - val_loss: 89803612160.0000 - val_mse: 898036
12160.0000 - val_mae: 255957.4531
Epoch 96/100
507/507 [==============================] - 2s 4ms/step - loss: 89087303680.0000 - ms
e: 89087303680.0000 - mae: 255587.9062 - val_loss: 89803137024.0000 - val_mse: 898031
37024.0000 - val_mae: 255956.4531
Epoch 97/100
507/507 [==============================] - 2s 5ms/step - loss: 89086779392.0000 - ms
e: 89086779392.0000 - mae: 255586.9375 - val_loss: 89802571776.0000 - val_mse: 898025
71776.0000 - val_mae: 255955.4062
Epoch 98/100
507/507 [==============================] - 3s 5ms/step - loss: 89086271488.0000 - ms
e: 89086271488.0000 - mae: 255585.9844 - val_loss: 89802104832.0000 - val_mse: 898021
04832.0000 - val_mae: 255954.4531
Epoch 99/100
507/507 [==============================] - 2s 4ms/step - loss: 89085739008.0000 - ms
e: 89085739008.0000 - mae: 255584.7969 - val_loss: 89801531392.0000 - val_mse: 898015
31392.0000 - val_mae: 255953.4219
Epoch 100/100
507/507 [==============================] - 2s 4ms/step - loss: 89085222912.0000 - ms
e: 89085222912.0000 - mae: 255583.9844 - val_loss: 89801064448.0000 - val_mse: 898010
64448.0000 - val_mae: 255952.4062
```

In [49]:
```
history_dict=history.history
```

In [50]:
```
history.history.keys()
```

Out[50]:
```
dict_keys(['loss', 'mse', 'mae', 'val_loss', 'val_mse', 'val_mae'])
```

In [51]:
```
history_dict.keys()
```

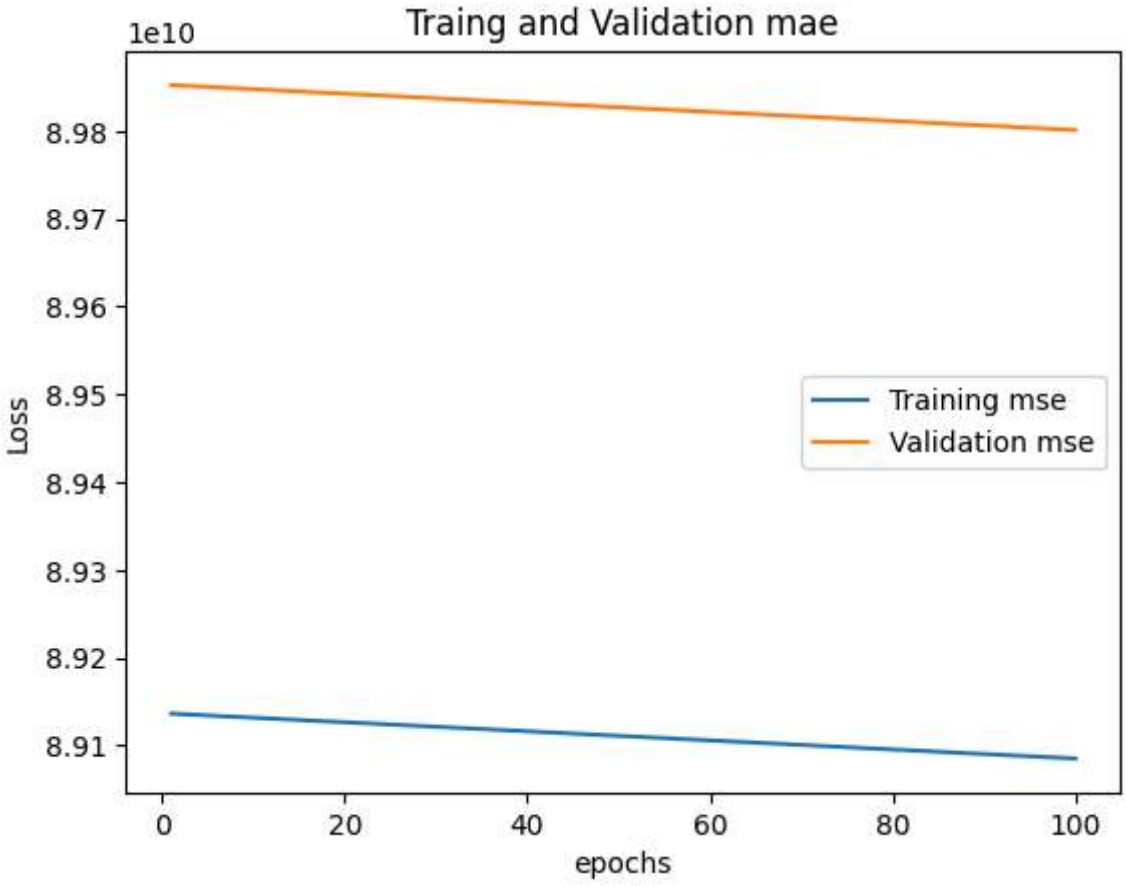
Out[51]:
```
dict_keys(['loss', 'mse', 'mae', 'val_loss', 'val_mse', 'val_mae'])
```

In [52]:
```
import matplotlib.pyplot as plt

mse_values=history_dict['mse']
val_mse_values=history_dict['val_mse']
epoch=range(1,len(mse_values)+1)

plt.plot(epoch,mse_values,label='Training mse')
```

```
plt.plot(epoch,val_mse_values,label="Validation mse")
plt.title("Traing and Validation mae")
plt.xlabel("epochs")
plt.ylabel("Loss")
#plt.ylim(0,100)
plt.legend()
plt.show()
```




I give up trying to get this to overfit it just won't overfit. It doesn't seem to overfit the performace seems to get better linearly, perhaps we could see overfitting if we ran for a day or so.

In [131…

```
!pip install shap
```

```
Requirement already satisfied: shap in /usr/local/lib/python3.10/dist-packages (0.44.
1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from
shap) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from
shap) (1.11.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-package
s (from shap) (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (fro
m shap) (1.5.3)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-package
s (from shap) (4.66.2)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packa
ges (from shap) (23.2)
Requirement already satisfied: slicer==0.0.7 in /usr/local/lib/python3.10/dist-packag
es (from shap) (0.0.7)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from
shap) (0.58.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages
(from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.1
0/dist-packages (from numba->shap) (0.41.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/di
st-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-package
s (from pandas->shap) (2023.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packag
es (from scikit-learn->shap) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist
-packages (from scikit-learn->shap) (3.3.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (f
rom python-dateutil>=2.8.1->pandas->shap) (1.16.0)
```

In [132…
```
import shap
```

In [132…
```

```

In [113…
```
explainer = shap.KernelExplainer(model, X_train2[:50,:])
```

In [114…
```
shap_values = explainer.shap_values(X_train2[20,:], nsamples=500)
```

```
divide by zero encountered in log
invalid value encountered in divide
```

In [116…
```
shap.force_plot(explainer.expected_value, shap_values[0], X_train2[20,:])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-116-35adeca3f794> in <cell line: 1>()
----> 1 shap.force_plot(explainer.expected_value, shap_values[0], X_train2[20,:])

/usr/local/lib/python3.10/dist-packages/shap/plots/_force.py in force(base_value, sha
p_values, features, feature_names, out_names, link, plot_cmap, matplotlib, show, figs
ize, ordering_keys, ordering_keys_time_format, text_rotation, contribution_threshold)
    128                 "shap.plots.force(explainer.expected_value[0], shap_values
[0])."
    129             )
--> 130             raise TypeError(emsg)
    131
    132     if isinstance(shap_values, list):

TypeError: In v0.20, force plot now requires the base value as the first parameter! T
ry shap.plots.force(explainer.expected_value, shap_values) or for multi-output models
try shap.plots.force(explainer.expected_value[0], shap_values[0]).
```

In [135…

```
!pip install Eli5
```

```
Requirement already satisfied: Eli5 in /usr/local/lib/python3.10/dist-packages (0.13.
0)
Requirement already satisfied: attrs>17.1.0 in /usr/local/lib/python3.10/dist-package
s (from Eli5) (23.2.0)
Requirement already satisfied: jinja2>=3.0.0 in /usr/local/lib/python3.10/dist-packag
es (from Eli5) (3.1.3)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.10/dist-package
s (from Eli5) (1.25.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from
Eli5) (1.11.4)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from E
li5) (1.16.0)
Requirement already satisfied: scikit-learn>=0.20 in /usr/local/lib/python3.10/dist-p
ackages (from Eli5) (1.2.2)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (f
rom Eli5) (0.20.1)
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.10/dist-pack
ages (from Eli5) (0.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-pack
ages (from jinja2>=3.0.0->Eli5) (2.1.5)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packag
es (from scikit-learn>=0.20->Eli5) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist
-packages (from scikit-learn>=0.20->Eli5) (3.3.0)
```

In [137…

```
perm = PermutationImportance(model, random_state=1, scoring="explained_variance").fit(
```

```
1013/1013 [==============================] - 2s 2ms/step
1013/1013 [==============================] - 1s 1ms/step
 976/1013 [===========================>..] - ETA: 0s
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-137-7ef0fd3fd507> in <cell line: 1>()
----> 1 perm = PermutationImportance(model, random_state=1, scoring="explained_varian
ce").fit(X_train2, y_train2)

/usr/local/lib/python3.10/dist-packages/eli5/sklearn/permutation_importance.py in fit
(self, X, y, groups, **fit_params)
    202             si = self._cv_scores_importances(X, y, groups=groups, **fit_param
s)
    203         else:
--> 204             si = self._non_cv_scores_importances(X, y)
    205         scores, results = si
    206         self.scores_ = np.array(scores)

/usr/local/lib/python3.10/dist-packages/eli5/sklearn/permutation_importance.py in _no
n_cv_scores_importances(self, X, y)
    230     def _non_cv_scores_importances(self, X, y):
    231         score_func = partial(self.scorer_, self.wrapped_estimator_)
--> 232         base_score, importances = self._get_score_importances(score_func, X,
y)
    233         return [base_score] * len(importances), importances
    234

/usr/local/lib/python3.10/dist-packages/eli5/sklearn/permutation_importance.py in _ge
t_score_importances(self, score_func, X, y)
    234
    235     def _get_score_importances(self, score_func, X, y):
--> 236         return get_score_importances(score_func, X, y, n_iter=self.n_iter,
    237                                      random_state=self.rng_)
    238

/usr/local/lib/python3.10/dist-packages/eli5/permutation_importance.py in get_score_i
mportances(score_func, X, y, n_iter, columns_to_shuffle, random_state)
     87     scores_decreases = []
     88     for i in range(n_iter):
---> 89         scores_shuffled = _get_scores_shufled(
     90             score_func, X, y, columns_to_shuffle=columns_to_shuffle,
     91             random_state=rng

/usr/local/lib/python3.10/dist-packages/eli5/permutation_importance.py in _get_scores
_shufled(score_func, X, y, columns_to_shuffle, random_state)
     98                           random_state=None):
     99     Xs = iter_shuffled(X, columns_to_shuffle, random_state=random_state)
--> 100     return np.array([score_func(X_shuffled, y) for X_shuffled in Xs])

/usr/local/lib/python3.10/dist-packages/eli5/permutation_importance.py in <listcomp>
(.0)
     98                           random_state=None):
     99     Xs = iter_shuffled(X, columns_to_shuffle, random_state=random_state)
--> 100     return np.array([score_func(X_shuffled, y) for X_shuffled in Xs])

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_scorer.py in __call__(self,
estimator, X, y_true, sample_weight)
    232             Score function applied to prediction of estimator on X.
    233         """
--> 234         return self._score(
    235             partial(_cached_call, None),
    236             estimator,
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_scorer.py in _score(self, method_caller, estimator, X, y_true, sample_weight)
    274         """
    275
--> 276         y_pred = method_caller(estimator, "predict", X)
    277         if sample_weight is not None:
    278             return self._sign * self._score_func(

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_scorer.py in _cached_call(cache, estimator, method, *args, **kwargs)
     71     """Call estimator with method and args and kwargs."""
     72     if cache is None:
---> 73         return getattr(estimator, method)(*args, **kwargs)
     74
     75     try:

/usr/local/lib/python3.10/dist-packages/keras/src/utils/traceback_utils.py in error_handler(*args, **kwargs)
     63         filtered_tb = None
     64         try:
---> 65             return fn(*args, **kwargs)
     66         except Exception as e:
     67             filtered_tb = _process_traceback_frames(e.__traceback__)

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py in predict(self, x, batch_size, verbose, steps, callbacks, max_queue_size, workers, use_multiprocessing)
   2653                     for step in data_handler.steps():
   2654                         callbacks.on_predict_batch_begin(step)
-> 2655                         tmp_batch_outputs = self.predict_function(iterator)
   2656                         if data_handler.should_sync:
   2657                             context.async_wait()

/usr/local/lib/python3.10/dist-packages/tensorflow/python/util/traceback_utils.py in error_handler(*args, **kwargs)
    148     filtered_tb = None
    149     try:
--> 150       return fn(*args, **kwargs)
    151     except Exception as e:
    152       filtered_tb = _process_traceback_frames(e.__traceback__)

/usr/local/lib/python3.10/dist-packages/tensorflow/python/eager/polymorphic_function/polymorphic_function.py in __call__(self, *args, **kwds)
    830
    831       with OptionalXlaContext(self._jit_compile):
--> 832         result = self._call(*args, **kwds)
    833
    834       new_tracing_count = self.experimental_get_tracing_count()

/usr/local/lib/python3.10/dist-packages/tensorflow/python/eager/polymorphic_function/polymorphic_function.py in _call(self, *args, **kwds)
    875       # In this case we have not created variables on the first call. So we can
    876       # run the first trace but we should fail if variables are created.
--> 877       results = tracing_compilation.call_function(
    878           args, kwds, self._variable_creation_config
    879       )

/usr/local/lib/python3.10/dist-packages/tensorflow/python/eager/polymorphic_function/tracing_compilation.py in call_function(args, kwargs, tracing_options)
```

```
    136   # Bind it ourselves to skip unnecessary canonicalization of default call.
    137   bound_args = function.function_type.bind(*args, **kwargs)
--> 138   flat_inputs = function.function_type.unpack_inputs(bound_args)
    139   return function._call_flat(  # pylint: disable=protected-access
    140       flat_inputs, captured_inputs=function.captured_inputs

/usr/local/lib/python3.10/dist-packages/tensorflow/core/function/polymorphism/function_type.py in unpack_inputs(self, bound_parameters)
    389     for p in sorted_parameters:
    390       flat.extend(
--> 391           p.type_constraint.to_tensors(bound_parameters.arguments[p.name])
    392       )
    393

/usr/local/lib/python3.10/dist-packages/tensorflow/python/framework/type_spec.py in to_tensors(self, value)
    252     nest.map_structure(
    253         lambda spec, v: tensors.extend(spec.to_tensors(v)),
--> 254         self._component_specs,
    255         self._to_components(value))
    256     return tensors

/usr/local/lib/python3.10/dist-packages/tensorflow/python/data/ops/iterator_ops.py in _component_specs(self)
    947   @property
    948   def _component_specs(self):
--> 949     return (tensor.TensorSpec([], dtypes.resource),)
    950
    951   def _to_components(self, value):

/usr/local/lib/python3.10/dist-packages/tensorflow/python/framework/tensor.py in __init__(self, shape, dtype, name)
    864         not convertible to a `tf.DType`.
    865     """
--> 866     self._shape = tensor_shape.TensorShape(shape)
    867     self._dtype = dtypes.as_dtype(dtype)
    868     self._name = name

/usr/local/lib/python3.10/dist-packages/tensorflow/python/framework/tensor_shape.py in __init__(self, dims)
    828     """
    829     if isinstance(dims, (tuple, list)):  # Most common case.
--> 830       self._dims = tuple(as_dimension(d).value for d in dims)
    831     elif dims is None:
    832       self._dims = None

KeyboardInterrupt:
```

In [53]:
```
#This takes a long time I give up trying to get it to work. I ran it for an hour.
```

# use L1, L2 and dropout to try to reduce overfitting, what method seems to work best for you?

l2_model = models.Sequential() l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(0.001), activation='relu', input_shape= (lyon_housing_final.shape[1],))) l2_model.add(layers.Dense(16, kernel_regularizer=regularizers.l2(0.001), activation='relu')) l2_model.add(layers.Dense(1, activation='sigmoid'))

In [60]:
```
from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras import regularizers
def build_model_l2():
    modelL2=models.Sequential()
    modelL2.add(layers.Dense(64,kernel_regularizer=regularizers.l2(0.001),activation='
    modelL2.add(layers.Dense(64,kernel_regularizer=regularizers.l2(0.001),activation='
    modelL2.add(layers.Dense(64,kernel_regularizer=regularizers.l2(0.001),activation='
    modelL2.add(layers.Dense(10,kernel_regularizer=regularizers.l2(0.001),activation='
    modelL2.add(layers.Dense(1))

    modelL2.compile(optimizer='rmsprop',loss='mse',metrics=['mse','mae'])
    return(modelL2)
```

In [61]:
```
model2 = build_model_l2()
# Train the model (in silent mode, verbose=0)
num_epochs=100
```

In [62]:
```
l2_model_hist = model2.fit(X_train2, y_train2,epochs=num_epochs, batch_size=64, verbos
```

```
Epoch 1/100
507/507 [==============================] - 4s 5ms/step - loss: 89136472064.0000 - ms
e: 89136472064.0000 - mae: 255683.9375 - val_loss: 89852370944.0000 - val_mse: 898523
70944.0000 - val_mae: 256052.6406
Epoch 2/100
507/507 [==============================] - 2s 4ms/step - loss: 89135898624.0000 - ms
e: 89135898624.0000 - mae: 255683.0781 - val_loss: 89851830272.0000 - val_mse: 898518
30272.0000 - val_mae: 256051.5781
Epoch 3/100
507/507 [==============================] - 2s 4ms/step - loss: 89135456256.0000 - ms
e: 89135456256.0000 - mae: 255682.0312 - val_loss: 89851314176.0000 - val_mse: 898513
14176.0000 - val_mae: 256050.6406
Epoch 4/100
507/507 [==============================] - 2s 5ms/step - loss: 89134882816.0000 - ms
e: 89134882816.0000 - mae: 255681.1094 - val_loss: 89850814464.0000 - val_mse: 898508
14464.0000 - val_mae: 256049.5938
Epoch 5/100
507/507 [==============================] - 3s 5ms/step - loss: 89134366720.0000 - ms
e: 89134366720.0000 - mae: 255680.0938 - val_loss: 89850298368.0000 - val_mse: 898502
98368.0000 - val_mae: 256048.5781
Epoch 6/100
507/507 [==============================] - 3s 5ms/step - loss: 89133801472.0000 - ms
e: 89133801472.0000 - mae: 255678.9062 - val_loss: 89849765888.0000 - val_mse: 898497
65888.0000 - val_mae: 256047.5312
Epoch 7/100
507/507 [==============================] - 2s 4ms/step - loss: 89133326336.0000 - ms
e: 89133326336.0000 - mae: 255678.0312 - val_loss: 89849282560.0000 - val_mse: 898492
82560.0000 - val_mae: 256046.6094
Epoch 8/100
507/507 [==============================] - 2s 4ms/step - loss: 89132818432.0000 - ms
e: 89132818432.0000 - mae: 255677.0938 - val_loss: 89848733696.0000 - val_mse: 898487
33696.0000 - val_mae: 256045.4844
Epoch 9/100
507/507 [==============================] - 2s 4ms/step - loss: 89132310528.0000 - ms
e: 89132310528.0000 - mae: 255676.0000 - val_loss: 89848225792.0000 - val_mse: 898482
25792.0000 - val_mae: 256044.5625
Epoch 10/100
507/507 [==============================] - 3s 5ms/step - loss: 89131810816.0000 - ms
e: 89131810816.0000 - mae: 255675.0156 - val_loss: 89847685120.0000 - val_mse: 898476
85120.0000 - val_mae: 256043.5000
Epoch 11/100
507/507 [==============================] - 3s 5ms/step - loss: 89131311104.0000 - ms
e: 89131311104.0000 - mae: 255674.0000 - val_loss: 89847185408.0000 - val_mse: 898471
85408.0000 - val_mae: 256042.5000
Epoch 12/100
507/507 [==============================] - 2s 4ms/step - loss: 89130745856.0000 - ms
e: 89130745856.0000 - mae: 255672.9531 - val_loss: 89846644736.0000 - val_mse: 898466
44736.0000 - val_mae: 256041.4688
Epoch 13/100
507/507 [==============================] - 2s 5ms/step - loss: 89130295296.0000 - ms
e: 89130295296.0000 - mae: 255672.0000 - val_loss: 89846145024.0000 - val_mse: 898461
45024.0000 - val_mae: 256040.5156
Epoch 14/100
507/507 [==============================] - 2s 4ms/step - loss: 89129762816.0000 - ms
e: 89129762816.0000 - mae: 255671.0312 - val_loss: 89845612544.0000 - val_mse: 898456
12544.0000 - val_mae: 256039.4688
Epoch 15/100
507/507 [==============================] - 3s 5ms/step - loss: 89129181184.0000 - ms
e: 89129181184.0000 - mae: 255669.9844 - val_loss: 89845112832.0000 - val_mse: 898451
12832.0000 - val_mae: 256038.4688
```

```
Epoch 16/100
507/507 [==============================] - 3s 5ms/step - loss: 89128697856.0000 - ms
e: 89128697856.0000 - mae: 255668.8750 - val_loss: 89844580352.0000 - val_mse: 898445
80352.0000 - val_mae: 256037.4531
Epoch 17/100
507/507 [==============================] - 4s 7ms/step - loss: 89128173568.0000 - ms
e: 89128173568.0000 - mae: 255667.8906 - val_loss: 89844088832.0000 - val_mse: 898440
88832.0000 - val_mae: 256036.4531
Epoch 18/100
507/507 [==============================] - 2s 4ms/step - loss: 89127690240.0000 - ms
e: 89127690240.0000 - mae: 255666.9062 - val_loss: 89843531776.0000 - val_mse: 898435
31776.0000 - val_mae: 256035.4062
Epoch 19/100
507/507 [==============================] - 2s 4ms/step - loss: 89127141376.0000 - ms
e: 89127141376.0000 - mae: 255665.8906 - val_loss: 89843048448.0000 - val_mse: 898430
48448.0000 - val_mae: 256034.4531
Epoch 20/100
507/507 [==============================] - 3s 6ms/step - loss: 89126592512.0000 - ms
e: 89126592512.0000 - mae: 255664.9219 - val_loss: 89842499584.0000 - val_mse: 898424
99584.0000 - val_mae: 256033.4219
Epoch 21/100
507/507 [==============================] - 3s 5ms/step - loss: 89126174720.0000 - ms
e: 89126174720.0000 - mae: 255663.8438 - val_loss: 89842008064.0000 - val_mse: 898420
08064.0000 - val_mae: 256032.4062
Epoch 22/100
507/507 [==============================] - 2s 4ms/step - loss: 89125593088.0000 - ms
e: 89125593088.0000 - mae: 255663.0156 - val_loss: 89841475584.0000 - val_mse: 898414
75584.0000 - val_mae: 256031.3906
Epoch 23/100
507/507 [==============================] - 2s 4ms/step - loss: 89125076992.0000 - ms
e: 89125076992.0000 - mae: 255662.0000 - val_loss: 89840967680.0000 - val_mse: 898409
67680.0000 - val_mae: 256030.4219
Epoch 24/100
507/507 [==============================] - 2s 5ms/step - loss: 89124503552.0000 - ms
e: 89124503552.0000 - mae: 255660.8281 - val_loss: 89840435200.0000 - val_mse: 898404
35200.0000 - val_mae: 256029.3750
Epoch 25/100
507/507 [==============================] - 3s 6ms/step - loss: 89124052992.0000 - ms
e: 89124052992.0000 - mae: 255659.8750 - val_loss: 89839927296.0000 - val_mse: 898399
27296.0000 - val_mae: 256028.4062
Epoch 26/100
507/507 [==============================] - 2s 4ms/step - loss: 89123577856.0000 - ms
e: 89123577856.0000 - mae: 255658.8594 - val_loss: 89839411200.0000 - val_mse: 898394
11200.0000 - val_mae: 256027.3750
Epoch 27/100
507/507 [==============================] - 2s 5ms/step - loss: 89122996224.0000 - ms
e: 89122996224.0000 - mae: 255657.8750 - val_loss: 89838895104.0000 - val_mse: 898388
95104.0000 - val_mae: 256026.3750
Epoch 28/100
507/507 [==============================] - 2s 5ms/step - loss: 89122480128.0000 - ms
e: 89122480128.0000 - mae: 255656.7344 - val_loss: 89838379008.0000 - val_mse: 898383
79008.0000 - val_mae: 256025.3125
Epoch 29/100
507/507 [==============================] - 2s 4ms/step - loss: 89121947648.0000 - ms
e: 89121947648.0000 - mae: 255655.9844 - val_loss: 89837879296.0000 - val_mse: 898378
79296.0000 - val_mae: 256024.3750
Epoch 30/100
507/507 [==============================] - 3s 6ms/step - loss: 89121480704.0000 - ms
e: 89121480704.0000 - mae: 255654.8438 - val_loss: 89837346816.0000 - val_mse: 898373
46816.0000 - val_mae: 256023.3125
```

```
Epoch 31/100
507/507 [==============================] - 2s 5ms/step - loss: 89121005568.0000 - ms
e: 89121005568.0000 - mae: 255653.7969 - val_loss: 89836806144.0000 - val_mse: 898368
06144.0000 - val_mae: 256022.3438
Epoch 32/100
507/507 [==============================] - 2s 4ms/step - loss: 89120423936.0000 - ms
e: 89120423936.0000 - mae: 255652.7500 - val_loss: 89836314624.0000 - val_mse: 898363
14624.0000 - val_mae: 256021.2656
Epoch 33/100
507/507 [==============================] - 2s 4ms/step - loss: 89119907840.0000 - ms
e: 89119907840.0000 - mae: 255651.8594 - val_loss: 89835790336.0000 - val_mse: 898357
90336.0000 - val_mae: 256020.3125
Epoch 34/100
507/507 [==============================] - 2s 4ms/step - loss: 89119391744.0000 - ms
e: 89119391744.0000 - mae: 255650.7812 - val_loss: 89835257856.0000 - val_mse: 898352
57856.0000 - val_mae: 256019.2188
Epoch 35/100
507/507 [==============================] - 3s 5ms/step - loss: 89118834688.0000 - ms
e: 89118834688.0000 - mae: 255649.7656 - val_loss: 89834758144.0000 - val_mse: 898347
58144.0000 - val_mae: 256018.2812
Epoch 36/100
507/507 [==============================] - 2s 5ms/step - loss: 89118359552.0000 - ms
e: 89118359552.0000 - mae: 255648.7031 - val_loss: 89834225664.0000 - val_mse: 898342
25664.0000 - val_mae: 256017.2188
Epoch 37/100
507/507 [==============================] - 2s 4ms/step - loss: 89117835264.0000 - ms
e: 89117835264.0000 - mae: 255647.7500 - val_loss: 89833717760.0000 - val_mse: 898337
17760.0000 - val_mae: 256016.2344
Epoch 38/100
507/507 [==============================] - 2s 4ms/step - loss: 89117343744.0000 - ms
e: 89117343744.0000 - mae: 255646.7344 - val_loss: 89833201664.0000 - val_mse: 898332
01664.0000 - val_mae: 256015.2031
Epoch 39/100
507/507 [==============================] - 2s 4ms/step - loss: 89116811264.0000 - ms
e: 89116811264.0000 - mae: 255645.7344 - val_loss: 89832685568.0000 - val_mse: 898326
85568.0000 - val_mae: 256014.2031
Epoch 40/100
507/507 [==============================] - 3s 5ms/step - loss: 89116352512.0000 - ms
e: 89116352512.0000 - mae: 255644.7656 - val_loss: 89832161280.0000 - val_mse: 898321
61280.0000 - val_mae: 256013.1562
Epoch 41/100
507/507 [==============================] - 3s 5ms/step - loss: 89115820032.0000 - ms
e: 89115820032.0000 - mae: 255643.6719 - val_loss: 89831628800.0000 - val_mse: 898316
28800.0000 - val_mae: 256012.1875
Epoch 42/100
507/507 [==============================] - 2s 4ms/step - loss: 89115222016.0000 - ms
e: 89115222016.0000 - mae: 255642.6406 - val_loss: 89831120896.0000 - val_mse: 898311
20896.0000 - val_mae: 256011.1719
Epoch 43/100
507/507 [==============================] - 2s 4ms/step - loss: 89114787840.0000 - ms
e: 89114787840.0000 - mae: 255641.5938 - val_loss: 89830563840.0000 - val_mse: 898305
63840.0000 - val_mae: 256010.1562
Epoch 44/100
507/507 [==============================] - 2s 4ms/step - loss: 89114255360.0000 - ms
e: 89114255360.0000 - mae: 255640.6094 - val_loss: 89830105088.0000 - val_mse: 898301
05088.0000 - val_mae: 256009.1094
Epoch 45/100
507/507 [==============================] - 3s 5ms/step - loss: 89113731072.0000 - ms
e: 89113731072.0000 - mae: 255639.6719 - val_loss: 89829556224.0000 - val_mse: 898295
56224.0000 - val_mae: 256008.1406
```

```
Epoch 46/100
507/507 [==============================] - 3s 5ms/step - loss: 89113223168.0000 - ms
e: 89113223168.0000 - mae: 255638.7031 - val_loss: 89829072896.0000 - val_mse: 898290
72896.0000 - val_mae: 256007.1250
Epoch 47/100
507/507 [==============================] - 2s 5ms/step - loss: 89112682496.0000 - ms
e: 89112682496.0000 - mae: 255637.5312 - val_loss: 89828548608.0000 - val_mse: 898285
48608.0000 - val_mae: 256006.1094
Epoch 48/100
507/507 [==============================] - 2s 5ms/step - loss: 89112182784.0000 - ms
e: 89112182784.0000 - mae: 255636.6094 - val_loss: 89828032512.0000 - val_mse: 898280
32512.0000 - val_mae: 256005.0625
Epoch 49/100
507/507 [==============================] - 2s 5ms/step - loss: 89111674880.0000 - ms
e: 89111674880.0000 - mae: 255635.7031 - val_loss: 89827516416.0000 - val_mse: 898275
16416.0000 - val_mae: 256004.0625
Epoch 50/100
507/507 [==============================] - 3s 5ms/step - loss: 89111158784.0000 - ms
e: 89111158784.0000 - mae: 255634.4688 - val_loss: 89827000320.0000 - val_mse: 898270
00320.0000 - val_mae: 256003.0469
Epoch 51/100
507/507 [==============================] - 3s 5ms/step - loss: 89110642688.0000 - ms
e: 89110642688.0000 - mae: 255633.6562 - val_loss: 89826476032.0000 - val_mse: 898264
76032.0000 - val_mae: 256002.0469
Epoch 52/100
507/507 [==============================] - 2s 4ms/step - loss: 89110126592.0000 - ms
e: 89110126592.0000 - mae: 255632.5000 - val_loss: 89825959936.0000 - val_mse: 898259
59936.0000 - val_mae: 256001.0312
Epoch 53/100
507/507 [==============================] - 2s 4ms/step - loss: 89109553152.0000 - ms
e: 89109553152.0000 - mae: 255631.5938 - val_loss: 89825443840.0000 - val_mse: 898254
43840.0000 - val_mae: 255999.9844
Epoch 54/100
507/507 [==============================] - 2s 4ms/step - loss: 89109110784.0000 - ms
e: 89109110784.0000 - mae: 255630.5156 - val_loss: 89824903168.0000 - val_mse: 898249
03168.0000 - val_mae: 255998.9688
Epoch 55/100
507/507 [==============================] - 3s 5ms/step - loss: 89108488192.0000 - ms
e: 89108488192.0000 - mae: 255629.5156 - val_loss: 89824411648.0000 - val_mse: 898244
11648.0000 - val_mae: 255997.9844
Epoch 56/100
507/507 [==============================] - 3s 5ms/step - loss: 89107980288.0000 - ms
e: 89107980288.0000 - mae: 255628.4688 - val_loss: 89823862784.0000 - val_mse: 898238
62784.0000 - val_mae: 255996.9219
Epoch 57/100
507/507 [==============================] - 2s 4ms/step - loss: 89107529728.0000 - ms
e: 89107529728.0000 - mae: 255627.4375 - val_loss: 89823363072.0000 - val_mse: 898233
63072.0000 - val_mae: 255995.9375
Epoch 58/100
507/507 [==============================] - 2s 5ms/step - loss: 89107013632.0000 - ms
e: 89107013632.0000 - mae: 255626.4531 - val_loss: 89822838784.0000 - val_mse: 898228
38784.0000 - val_mae: 255994.9531
Epoch 59/100
507/507 [==============================] - 2s 4ms/step - loss: 89106481152.0000 - ms
e: 89106481152.0000 - mae: 255625.3906 - val_loss: 89822314496.0000 - val_mse: 898223
14496.0000 - val_mae: 255993.9062
Epoch 60/100
507/507 [==============================] - 3s 5ms/step - loss: 89106014208.0000 - ms
e: 89106014208.0000 - mae: 255624.5156 - val_loss: 89821814784.0000 - val_mse: 898218
14784.0000 - val_mae: 255992.9062
```

```
Epoch 61/100
507/507 [==============================] - 3s 5ms/step - loss: 89105408000.0000 - ms
e: 89105408000.0000 - mae: 255623.4688 - val_loss: 89821290496.0000 - val_mse: 898212
90496.0000 - val_mae: 255991.9219
Epoch 62/100
507/507 [==============================] - 2s 4ms/step - loss: 89104957440.0000 - ms
e: 89104957440.0000 - mae: 255622.3594 - val_loss: 89820782592.0000 - val_mse: 898207
82592.0000 - val_mae: 255990.8906
Epoch 63/100
507/507 [==============================] - 2s 5ms/step - loss: 89104433152.0000 - ms
e: 89104433152.0000 - mae: 255621.3750 - val_loss: 89820241920.0000 - val_mse: 898202
41920.0000 - val_mae: 255989.8906
Epoch 64/100
507/507 [==============================] - 2s 4ms/step - loss: 89103941632.0000 - ms
e: 89103941632.0000 - mae: 255620.3906 - val_loss: 89819750400.0000 - val_mse: 898197
50400.0000 - val_mae: 255988.8906
Epoch 65/100
507/507 [==============================] - 2s 5ms/step - loss: 89103368192.0000 - ms
e: 89103368192.0000 - mae: 255619.4531 - val_loss: 89819209728.0000 - val_mse: 898192
09728.0000 - val_mae: 255987.8906
Epoch 66/100
507/507 [==============================] - 3s 6ms/step - loss: 89102843904.0000 - ms
e: 89102843904.0000 - mae: 255618.3438 - val_loss: 89818710016.0000 - val_mse: 898187
10016.0000 - val_mae: 255986.8438
Epoch 67/100
507/507 [==============================] - 2s 4ms/step - loss: 89102360576.0000 - ms
e: 89102360576.0000 - mae: 255617.3438 - val_loss: 89818185728.0000 - val_mse: 898181
85728.0000 - val_mae: 255985.8750
Epoch 68/100
507/507 [==============================] - 2s 5ms/step - loss: 89101885440.0000 - ms
e: 89101885440.0000 - mae: 255616.3438 - val_loss: 89817669632.0000 - val_mse: 898176
69632.0000 - val_mae: 255984.8594
Epoch 69/100
507/507 [==============================] - 2s 4ms/step - loss: 89101287424.0000 - ms
e: 89101287424.0000 - mae: 255615.2656 - val_loss: 89817153536.0000 - val_mse: 898171
53536.0000 - val_mae: 255983.8594
Epoch 70/100
507/507 [==============================] - 2s 5ms/step - loss: 89100804096.0000 - ms
e: 89100804096.0000 - mae: 255614.4062 - val_loss: 89816645632.0000 - val_mse: 898166
45632.0000 - val_mae: 255982.8438
Epoch 71/100
507/507 [==============================] - 3s 6ms/step - loss: 89100304384.0000 - ms
e: 89100304384.0000 - mae: 255613.3281 - val_loss: 89816137728.0000 - val_mse: 898161
37728.0000 - val_mae: 255981.8594
Epoch 72/100
507/507 [==============================] - 2s 4ms/step - loss: 89099755520.0000 - ms
e: 89099755520.0000 - mae: 255612.2812 - val_loss: 89815621632.0000 - val_mse: 898156
21632.0000 - val_mae: 255980.7969
Epoch 73/100
507/507 [==============================] - 2s 4ms/step - loss: 89099206656.0000 - ms
e: 89099206656.0000 - mae: 255611.3281 - val_loss: 89815064576.0000 - val_mse: 898150
64576.0000 - val_mae: 255979.7812
Epoch 74/100
507/507 [==============================] - 2s 4ms/step - loss: 89098674176.0000 - ms
e: 89098674176.0000 - mae: 255610.2344 - val_loss: 89814556672.0000 - val_mse: 898145
56672.0000 - val_mae: 255978.7969
Epoch 75/100
507/507 [==============================] - 2s 4ms/step - loss: 89098190848.0000 - ms
e: 89098190848.0000 - mae: 255609.2031 - val_loss: 89814024192.0000 - val_mse: 898140
24192.0000 - val_mae: 255977.7812
```

```
Epoch 76/100
507/507 [==============================] - 3s 6ms/step - loss: 89097707520.0000 - ms
e: 89097707520.0000 - mae: 255608.2500 - val_loss: 89813549056.0000 - val_mse: 898135
49056.0000 - val_mae: 255976.7969
Epoch 77/100
507/507 [==============================] - 2s 5ms/step - loss: 89097175040.0000 - ms
e: 89097175040.0000 - mae: 255607.2812 - val_loss: 89812992000.0000 - val_mse: 898129
92000.0000 - val_mae: 255975.7656
Epoch 78/100
507/507 [==============================] - 2s 4ms/step - loss: 89096675328.0000 - ms
e: 89096675328.0000 - mae: 255606.2812 - val_loss: 89812516864.0000 - val_mse: 898125
16864.0000 - val_mae: 255974.7656
Epoch 79/100
507/507 [==============================] - 2s 4ms/step - loss: 89096101888.0000 - ms
e: 89096101888.0000 - mae: 255605.3281 - val_loss: 89811968000.0000 - val_mse: 898119
68000.0000 - val_mae: 255973.7188
Epoch 80/100
507/507 [==============================] - 2s 4ms/step - loss: 89095610368.0000 - ms
e: 89095610368.0000 - mae: 255604.1094 - val_loss: 89811476480.0000 - val_mse: 898114
76480.0000 - val_mae: 255972.7500
Epoch 81/100
507/507 [==============================] - 3s 6ms/step - loss: 89095135232.0000 - ms
e: 89095135232.0000 - mae: 255603.1250 - val_loss: 89810919424.0000 - val_mse: 898109
19424.0000 - val_mae: 255971.7188
Epoch 82/100
507/507 [==============================] - 2s 5ms/step - loss: 89094586368.0000 - ms
e: 89094586368.0000 - mae: 255602.0781 - val_loss: 89810436096.0000 - val_mse: 898104
36096.0000 - val_mae: 255970.7500
Epoch 83/100
507/507 [==============================] - 2s 5ms/step - loss: 89094045696.0000 - ms
e: 89094045696.0000 - mae: 255601.0781 - val_loss: 89809895424.0000 - val_mse: 898098
95424.0000 - val_mae: 255969.6719
Epoch 84/100
507/507 [==============================] - 2s 5ms/step - loss: 89093554176.0000 - ms
e: 89093554176.0000 - mae: 255600.2812 - val_loss: 89809395712.0000 - val_mse: 898093
95712.0000 - val_mae: 255968.7188
Epoch 85/100
507/507 [==============================] - 2s 5ms/step - loss: 89093087232.0000 - ms
e: 89093087232.0000 - mae: 255599.0469 - val_loss: 89808863232.0000 - val_mse: 898088
63232.0000 - val_mae: 255967.6719
Epoch 86/100
507/507 [==============================] - 3s 6ms/step - loss: 89092513792.0000 - ms
e: 89092513792.0000 - mae: 255598.1875 - val_loss: 89808379904.0000 - val_mse: 898083
79904.0000 - val_mae: 255966.7188
Epoch 87/100
507/507 [==============================] - 2s 5ms/step - loss: 89091989504.0000 - ms
e: 89091989504.0000 - mae: 255597.1250 - val_loss: 89807814656.0000 - val_mse: 898078
14656.0000 - val_mae: 255965.6562
Epoch 88/100
507/507 [==============================] - 2s 5ms/step - loss: 89091563520.0000 - ms
e: 89091563520.0000 - mae: 255596.1094 - val_loss: 89807339520.0000 - val_mse: 898073
39520.0000 - val_mae: 255964.7031
Epoch 89/100
507/507 [==============================] - 2s 5ms/step - loss: 89090981888.0000 - ms
e: 89090981888.0000 - mae: 255595.1094 - val_loss: 89806782464.0000 - val_mse: 898067
82464.0000 - val_mae: 255963.6406
Epoch 90/100
507/507 [==============================] - 2s 4ms/step - loss: 89090400256.0000 - ms
e: 89090400256.0000 - mae: 255594.0312 - val_loss: 89806307328.0000 - val_mse: 898063
07328.0000 - val_mae: 255962.6875
```

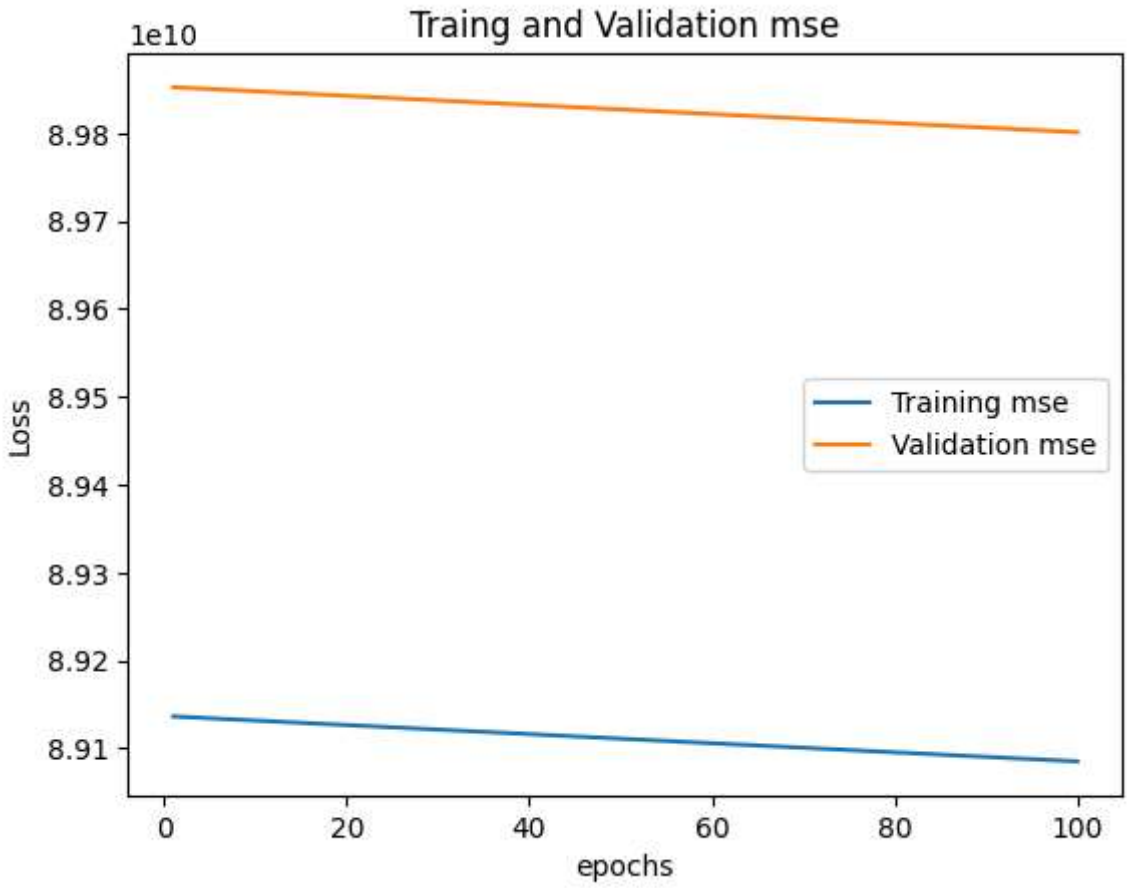
```
Epoch 91/100
507/507 [==============================] - 3s 6ms/step - loss: 89089957888.0000 - ms
e: 89089957888.0000 - mae: 255593.1094 - val_loss: 89805742080.0000 - val_mse: 898057
42080.0000 - val_mae: 255961.6094
Epoch 92/100
507/507 [==============================] - 2s 5ms/step - loss: 89089417216.0000 - ms
e: 89089417216.0000 - mae: 255592.0312 - val_loss: 89805258752.0000 - val_mse: 898052
58752.0000 - val_mae: 255960.6406
Epoch 93/100
507/507 [==============================] - 2s 4ms/step - loss: 89088901120.0000 - ms
e: 89088901120.0000 - mae: 255591.0469 - val_loss: 89804726272.0000 - val_mse: 898047
26272.0000 - val_mae: 255959.5938
Epoch 94/100
507/507 [==============================] - 2s 5ms/step - loss: 89088442368.0000 - ms
e: 89088442368.0000 - mae: 255590.0781 - val_loss: 89804226560.0000 - val_mse: 898042
26560.0000 - val_mae: 255958.6406
Epoch 95/100
507/507 [==============================] - 2s 5ms/step - loss: 89087885312.0000 - ms
e: 89087885312.0000 - mae: 255589.0938 - val_loss: 89803669504.0000 - val_mse: 898036
69504.0000 - val_mae: 255957.5938
Epoch 96/100
507/507 [==============================] - 3s 6ms/step - loss: 89087385600.0000 - ms
e: 89087385600.0000 - mae: 255588.1406 - val_loss: 89803194368.0000 - val_mse: 898031
94368.0000 - val_mae: 255956.5938
Epoch 97/100
507/507 [==============================] - 2s 5ms/step - loss: 89086787584.0000 - ms
e: 89086787584.0000 - mae: 255587.1406 - val_loss: 89802637312.0000 - val_mse: 898026
37312.0000 - val_mae: 255955.5156
Epoch 98/100
507/507 [==============================] - 2s 4ms/step - loss: 89086353408.0000 - ms
e: 89086353408.0000 - mae: 255585.9844 - val_loss: 89802153984.0000 - val_mse: 898021
53984.0000 - val_mae: 255954.6094
Epoch 99/100
507/507 [==============================] - 2s 5ms/step - loss: 89085812736.0000 - ms
e: 89085812736.0000 - mae: 255585.0312 - val_loss: 89801596928.0000 - val_mse: 898015
96928.0000 - val_mae: 255953.5000
Epoch 100/100
507/507 [==============================] - 2s 5ms/step - loss: 89085280256.0000 - ms
e: 89085280256.0000 - mae: 255583.9688 - val_loss: 89801121792.0000 - val_mse: 898011
21792.0000 - val_mae: 255952.5156
```

In [64]:

```python
import matplotlib.pyplot as plt

mse_values=history_dict['mse']
val_mse_values=history_dict['val_mse']
epoch=range(1,len(mse_values)+1)

plt.plot(epoch,mse_values,label='Training mse')
plt.plot(epoch,val_mse_values,label="Validation mse")
plt.title("Traing and Validation mse")
plt.xlabel("epochs")
plt.ylabel("Loss")
#plt.ylim(0,100)
plt.legend()
plt.show()
```

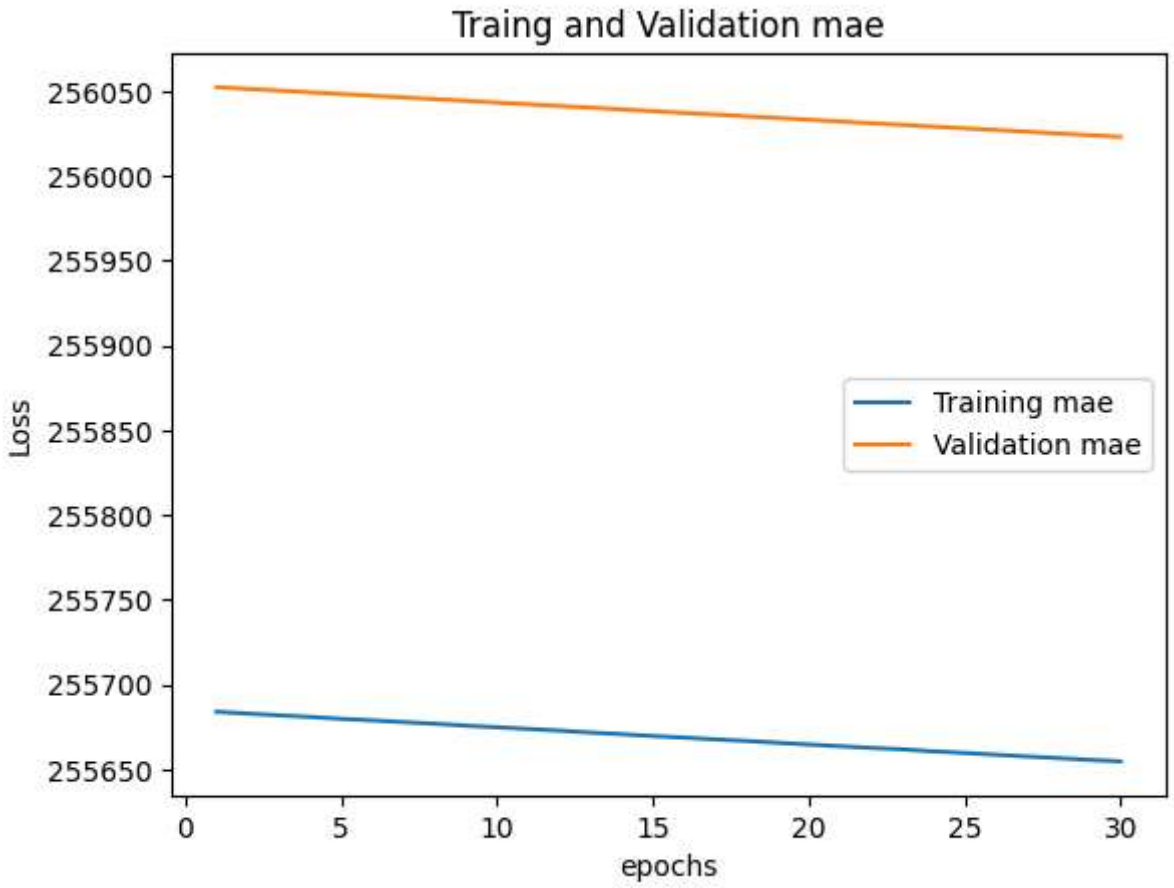



In [163…
```
history_dict2=l2_model_hist.history
```

In [164…
```
import matplotlib.pyplot as plt

mse_values=history_dict2['mae']
val_mse_values=history_dict2['val_mae']
epoch=range(1,len(mse_values)+1)

plt.plot(epoch,mse_values,label='Training mae')
plt.plot(epoch,val_mse_values,label="Validation mae")
plt.title("Traing and Validation mae")
plt.xlabel("epochs")
plt.ylabel("Loss")
#plt.ylim(0,100)
plt.legend()
plt.show()
```




In [66]:
```
#I will have to run this for a long time to do this assignment. But the mse seems to b
```

In [67]:
```
# It also takes a long time to get shap and Eli5 to run on this. I ran it for an hour
```

In [ ]:
```

```