houseprice-1

October 5, 2024

Project: House Price Prediction Dataset: Housing prices dataset Task: Predict house prices based on features like size, number of bedrooms, location

[ ]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

[ ]:
```
df= pd.read_csv('/content/House Price India.csv')
df
```

[ ]:
```
               id   Date  number of bedrooms  number of bathrooms  \
0      6762810635  42491                   4                 2.50
1      6762810998  42491                   5                 2.75
2      6762812605  42491                   4                 2.50
3      6762812919  42491                   3                 2.00
4      6762813105  42491                   3                 2.50
…             …     …                  …                   …
14614  6762830250  42734                   2                 1.50
14615  6762830339  42734                   3                 2.00
14616  6762830618  42734                   2                 1.00
14617  6762830709  42734                   4                 1.00
14618  6762831463  42734                   3                 1.00

       living area  lot area  number of floors  waterfront present  \
0             2920      4000               1.5                   0
1             2910      9480               1.5                   0
2             3310     42998               2.0                   0
3             2710      4500               1.5                   0
4             2600      4750               1.0                   0
…              …        …                 …                   …
14614         1556     20000               1.0                   0
14615         1680      7000               1.5                   0
14616         1070      6120               1.0                   0
14617         1030      6621               1.0                   0
14618          900      4770               1.0                   0
```

	number of views	condition of the house	…	Built Year	\
0	0	5	…	1909	
1	0	3	…	1939	
2	0	3	…	2001	
3	0	4	…	1929	
4	0	4	…	1951	
…	…	…	…	…	
14614	0	4	…	1957	
14615	0	4	…	1968	
14616	0	3	…	1962	
14617	0	4	…	1955	
14618	0	3	…	1969	

	Renovation Year	Postal Code	Lattitude	Longitude	living_area_renov	\
0	0	122004	52.8878	-114.470	2470	
1	0	122004	52.8852	-114.468	2940	
2	0	122005	52.9532	-114.321	3350	
3	0	122006	52.9047	-114.485	2060	
4	0	122007	52.9133	-114.590	2380	
…	…	…	…	…	…	
14614	0	122066	52.6191	-114.472	2250	
14615	0	122072	52.5075	-114.393	1540	
14616	0	122056	52.7289	-114.507	1130	
14617	0	122042	52.7157	-114.411	1420	
14618	2009	122018	52.5338	-114.552	900	

	lot_area_renov	Number of schools nearby	Distance from the airport	\
0	4000	2	51	
1	6600	1	53	
2	42847	3	76	
3	4500	1	51	
4	4750	1	67	
…	…	…	…	
14614	17286	3	76	
14615	7480	3	59	
14616	6120	2	64	
14617	6631	3	54	
14618	3480	2	55	

	Price
0	1400000
1	1200000
2	838000
3	805000
4	790000
…	…
14614	221700

```
14615    219200
14616    209000
14617    205000
14618    146000

[14619 rows x 23 columns]
```

[ ]:
```
df.shape
```

[ ]:
```
(14619, 23)
```

[ ]:
```
df.duplicated().sum()
```

[ ]:
```
0
```

[ ]:
```
!pip install  ydata_profiling
```

```
Collecting ydata_profiling
  Downloading ydata_profiling-4.10.0-py2.py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: scipy<1.14,>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (1.13.1)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (2.1.4)
Requirement already satisfied: matplotlib<3.10,>=3.5 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (3.7.1)
Requirement already satisfied: pydantic>=2 in /usr/local/lib/python3.10/dist-
packages (from ydata_profiling) (2.8.2)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (6.0.2)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (3.1.4)
Collecting visions<0.7.7,>=0.7.5 (from
visions[type_image_path]<0.7.7,>=0.7.5->ydata_profiling)
  Downloading visions-0.7.6-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.2,>=1.16.0 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (1.26.4)
Collecting htmlmin==0.1.12 (from ydata_profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) … done
Collecting phik<0.13,>=0.11.1 (from ydata_profiling)
  Downloading
phik-0.12.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(5.6 kB)
Requirement already satisfied: requests<3,>=2.24.0 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (2.32.3)
Requirement already satisfied: tqdm<5,>=4.48.2 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (4.66.5)
```

```
Requirement already satisfied: seaborn<0.14,>=0.10.1 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (0.13.1)
Collecting multimethod<2,>=1.4 (from ydata_profiling)
  Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (0.14.2)
Requirement already satisfied: typeguard<5,>=3 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (4.3.0)
Collecting imagehash==4.3.1 (from ydata_profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (1.9.3)
Collecting dacite>=1.8 (from ydata_profiling)
  Downloading dacite-1.8.1-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: numba<1,>=0.56.0 in
/usr/local/lib/python3.10/dist-packages (from ydata_profiling) (0.60.0)
Collecting PyWavelets (from imagehash==4.3.1->ydata_profiling)
  Downloading pywavelets-1.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl.metadata (9.0 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages
(from imagehash==4.3.1->ydata_profiling) (9.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from
jinja2<3.2,>=2.11.1->ydata_profiling) (2.1.5)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib<3.10,>=3.5->ydata_profiling) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib<3.10,>=3.5->ydata_profiling) (2.8.2)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
/usr/local/lib/python3.10/dist-packages (from numba<1,>=0.56.0->ydata_profiling)
(0.43.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
```

```
packages (from pandas!=1.4.0,<3,>1.1->ydata_profiling) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas!=1.4.0,<3,>1.1->ydata_profiling) (2024.1)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-
packages (from phik<0.13,>=0.11.1->ydata_profiling) (1.4.2)
Requirement already satisfied: annotated-types>=0.4.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata_profiling)
(0.7.0)
Requirement already satisfied: pydantic-core==2.20.1 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata_profiling)
(2.20.1)
Requirement already satisfied: typing-extensions>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata_profiling)
(4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.24.0->ydata_profiling) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.24.0->ydata_profiling) (3.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.24.0->ydata_profiling) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3,>=2.24.0->ydata_profiling) (2024.8.30)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-
packages (from statsmodels<1,>=0.13.2->ydata_profiling) (0.5.6)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.10/dist-
packages (from
visions<0.7.7,>=0.7.5->visions[type_image_path]<0.7.7,>=0.7.5->ydata_profiling)
(24.2.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.10/dist-
packages (from
visions<0.7.7,>=0.7.5->visions[type_image_path]<0.7.7,>=0.7.5->ydata_profiling)
(3.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.6->statsmodels<1,>=0.13.2->ydata_profiling) (1.16.0)
Downloading ydata_profiling-4.10.0-py2.py3-none-any.whl (356 kB)
                         356.2/356.2 kB
6.8 MB/s eta 0:00:00
Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
                         296.5/296.5 kB
14.2 MB/s eta 0:00:00
Downloading dacite-1.8.1-py3-none-any.whl (14 kB)
Downloading multimethod-1.12-py3-none-any.whl (10 kB)
Downloading
phik-0.12.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (686 kB)
                         686.1/686.1 kB
```

```
30.6 MB/s eta 0:00:00
Downloading visions-0.7.6-py3-none-any.whl (104 kB)
                         104.8/104.8 kB
5.2 MB/s eta 0:00:00
Downloading
pywavelets-1.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.5
MB)
                         4.5/4.5 MB
47.6 MB/s eta 0:00:00
Building wheels for collected packages: htmlmin
  Building wheel for htmlmin (setup.py) … done
  Created wheel for htmlmin: filename=htmlmin-0.1.12-py3-none-any.whl size=27081
sha256=a5e1df2ff69837e9d27b1a986db884138b521b9d0c3c370719790aa81898ffba
  Stored in directory: /root/.cache/pip/wheels/dd/91/29/a79cecb328d01739e64017b6
fb9a1ab9d8cb1853098ec5966d
Successfully built htmlmin
Installing collected packages: htmlmin, PyWavelets, multimethod, dacite,
imagehash, visions, phik, ydata_profiling
Successfully installed PyWavelets-1.7.0 dacite-1.8.1 htmlmin-0.1.12
imagehash-4.3.1 multimethod-1.12 phik-0.12.4 visions-0.7.6
ydata_profiling-4.10.0
```

[ ]:
```python
import ydata_profiling as pp
```

[ ]:
```python
data = pd.read_csv('/content/House Price India.csv')
pp.ProfileReport(data)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[ ]:

[ ]:
```python
df.describe()
```

[ ]:

	id	Date	number of bedrooms	number of bathrooms	\
count	1.461900e+04	14619.000000	14619.000000	14619.000000	
mean	6.762821e+09	42604.546412	3.379233	2.129557	
std	6.237162e+03	67.343747	0.938655	0.769955	
min	6.762810e+09	42491.000000	1.000000	0.500000	
25%	6.762815e+09	42546.000000	3.000000	1.750000	
50%	6.762821e+09	42600.000000	3.000000	2.250000	
75%	6.762826e+09	42662.000000	4.000000	2.500000	
max	6.762832e+09	42734.000000	33.000000	8.000000	

	living area	lot area	number of floors	waterfront present	\
count	14619.000000	1.461900e+04	14619.000000	14619.000000	
mean	2098.156851	1.509369e+04	1.502326	0.007661	
std	928.218740	3.792089e+04	0.540241	0.087196	
min	370.000000	5.200000e+02	1.000000	0.000000	
25%	1440.000000	5.010500e+03	1.000000	0.000000	
50%	1930.000000	7.620000e+03	1.500000	0.000000	
75%	2570.000000	1.080000e+04	2.000000	0.000000	
max	13540.000000	1.074218e+06	3.500000	1.000000	

	number of views	condition of the house	…	Built Year	\
count	14619.000000	14619.000000	…	14619.000000	
mean	0.232848	3.430399	…	1970.929817	
std	0.765651	0.664047	…	29.491743	
min	0.000000	1.000000	…	1900.000000	
25%	0.000000	3.000000	…	1951.000000	
50%	0.000000	3.000000	…	1975.000000	
75%	0.000000	4.000000	…	1997.000000	
max	4.000000	5.000000	…	2015.000000	

	Renovation Year	Postal Code	Lattitude	Longitude	\
count	14619.000000	14619.000000	14619.000000	14619.000000	
mean	90.930228	122033.064300	52.792843	-114.403996	
std	416.230218	19.081451	0.137525	0.141325	
min	0.000000	122003.000000	52.385900	-114.709000	
25%	0.000000	122017.000000	52.707600	-114.519000	
50%	0.000000	122032.000000	52.806400	-114.421000	
75%	0.000000	122048.000000	52.908900	-114.315000	
max	2015.000000	122072.000000	53.007600	-113.505000	

	living_area_renov	lot_area_renov	Number of schools nearby	\
count	14619.000000	14619.000000	14619.000000	
mean	1996.641836	12754.003078	2.012244	
std	691.078387	26059.234785	0.817312	
min	460.000000	651.000000	1.000000	
25%	1490.000000	5097.500000	1.000000	
50%	1850.000000	7620.000000	2.000000	
75%	2380.000000	10125.000000	3.000000	
max	6110.000000	560617.000000	3.000000	

	Distance from the airport	Price
count	14619.000000	1.461900e+04
mean	64.951433	5.388063e+05
std	8.936129	3.672294e+05
min	50.000000	7.800000e+04
25%	57.000000	3.200000e+05
50%	65.000000	4.500000e+05

```
75%                     73.000000  6.450000e+05
max                     80.000000  7.700000e+06

[8 rows x 23 columns]
```

[ ]:
```python
df_cat = df.select_dtypes(include='object')
df_cat
```

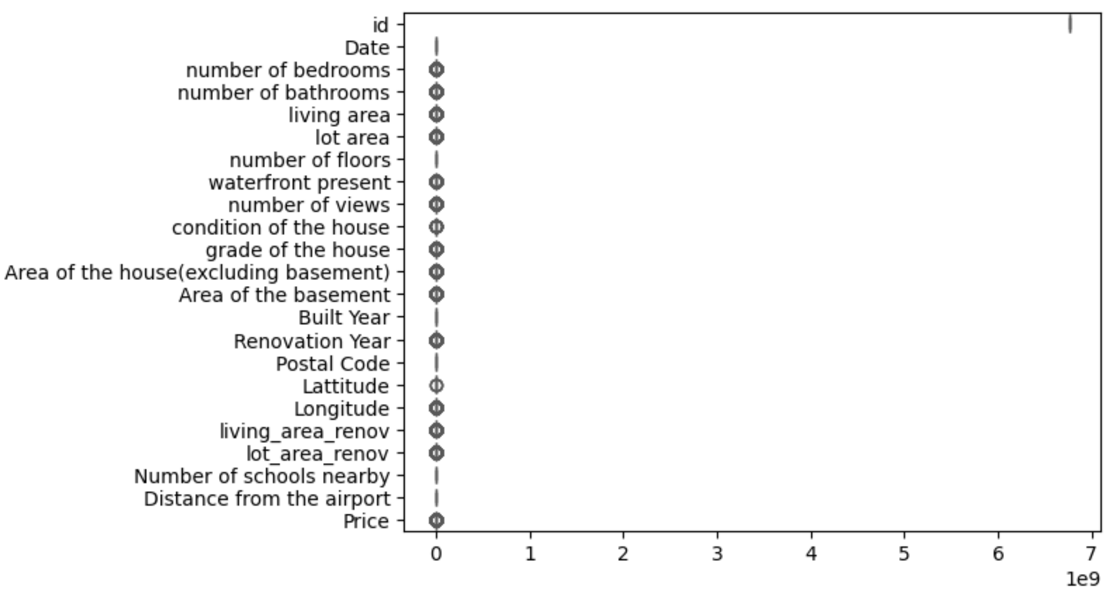
[ ]:
```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,
…]

[14619 rows x 0 columns]
```

[ ]:
```python
df.shape
```

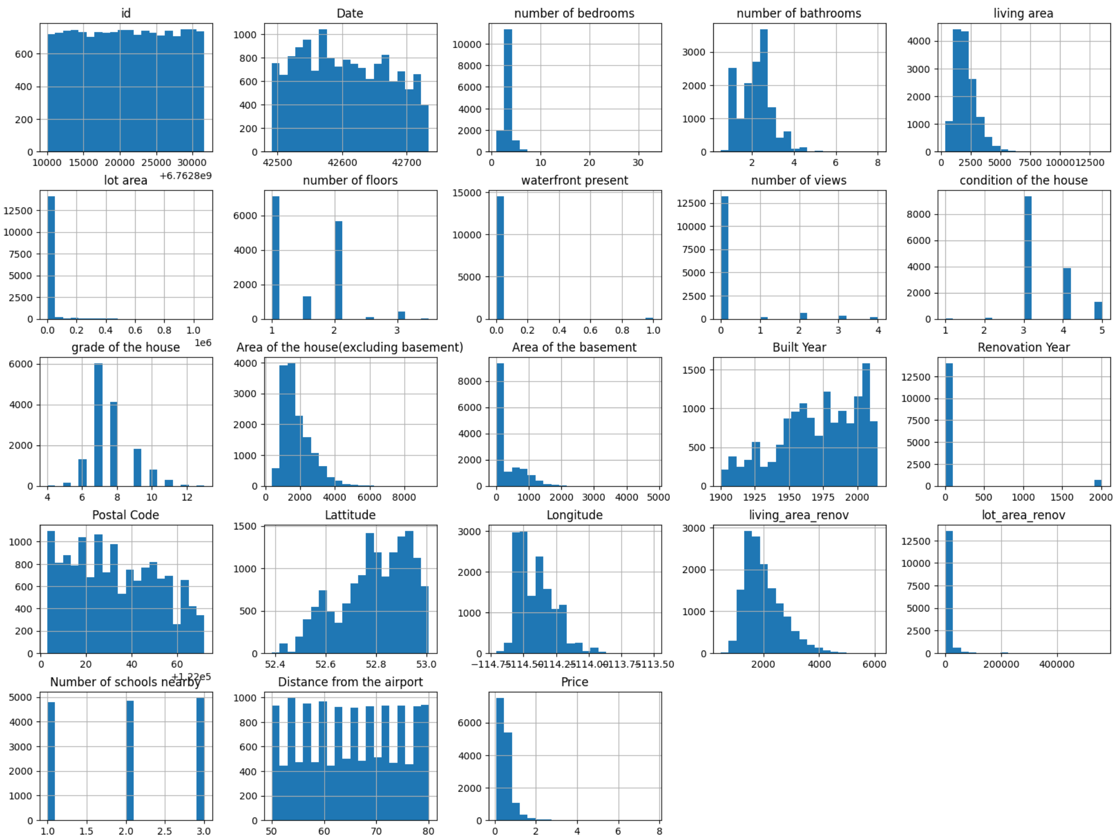
[ ]:
```
(525, 13)
```

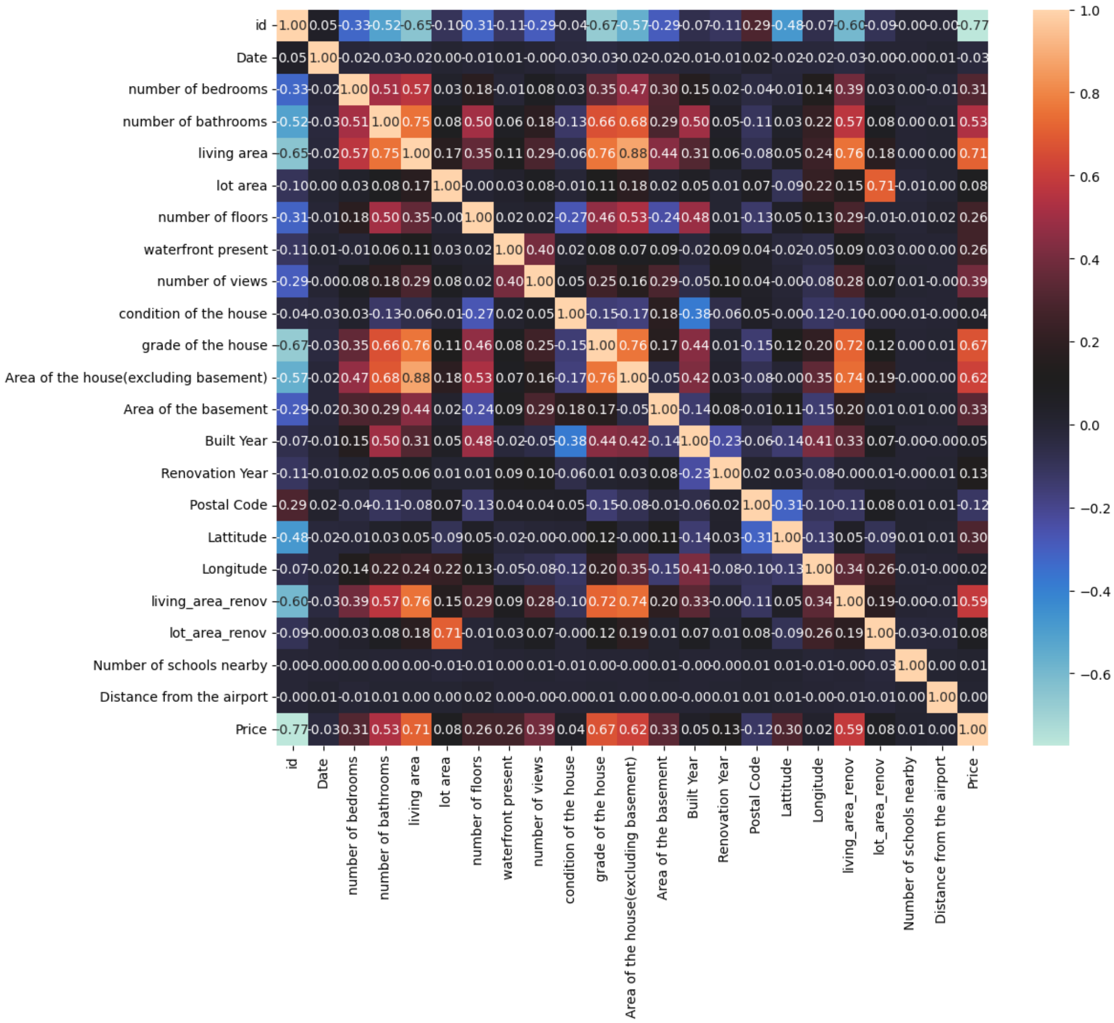
[ ]:
```python
import seaborn as sns
sns.boxplot(data=df, orient="h", palette="Set2", dodge=False)
```

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



```
[ ]: # Plotting distributions of numeric features
df.hist(bins=20, figsize=(20, 15))
plt.show()

# Correlation matrix
# Correlation matrix for numeric columns
numeric_columns = df.select_dtypes(include=[np.number])
corr_matrix = numeric_columns.corr()

# Plotting the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, fmt=".2f",cmap="icefire")
plt.show()
```





[ ]:
```
import math
```

[ ]:
```
cf =␣
↪['mainroad','guestroom',        'basement',        'hotwaterheating','airconditioning','pre
```

[ ]:
```
# Import library for VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor

def calc_vif(X):

    # Calculating VIF
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.
↪shape[1])]
```

```
    return(vif)
```

[ ]:
```
calc_vif(X)
```

```
/usr/local/lib/python3.10/dist-
packages/statsmodels/stats/outliers_influence.py:197: RuntimeWarning: divide by
zero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
/usr/local/lib/python3.10/dist-packages/pandas/core/nanops.py:1010:
RuntimeWarning: invalid value encountered in subtract
  sqr = _ensure_numeric((avg - values) ** 2)
```

[ ]:

	variables	VIF
0	id	4.913244e+07
1	Date	1.004228e+00
2	number of bedrooms	1.624363e+00
3	number of bathrooms	3.335665e+00
4	living area	inf
5	lot area	2.031741e+00
6	number of floors	2.005297e+00
7	waterfront present	1.202951e+00
8	number of views	1.439942e+00
9	condition of the house	1.262652e+00
10	grade of the house	3.459434e+00
11	Area of the house(excluding basement)	inf
12	Area of the basement	inf
13	Built Year	2.391590e+00
14	Renovation Year	1.159906e+00
15	Postal Code	1.166669e+00
16	Lattitude	1.224041e+00
17	Longitude	1.532602e+00
18	living_area_renov	3.026162e+00
19	lot_area_renov	2.080642e+00
20	Number of schools nearby	1.001518e+00
21	Distance from the airport	1.001856e+00

Model fitting

[ ]:
```
X=df.drop('Price',axis=1)
y=df.Price
```

[ ]:
```
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
 ↪30,random_state=70)
X_train.shape, X_test.shape
```

[ ]: ((10233, 22), (4386, 22))

Linear Regression

```
[ ]: from sklearn.linear_model import LinearRegression
     import matplotlib.pyplot as plt
```

```
[ ]: # Create a linear regression model
     lr = LinearRegression()
     # Fit the model to the data
     lr.fit(X_train, y_train)
     # Make predictions
     y_pred = lr.predict(X)
     # Print the predicted values for new data
     print("Predicted values for new data:",y_pred )
```

```
Predicted values for new data: [969672.35571289 894062.35873413 878718.20098877
… 133324.49261475
  50950.61312866  -1850.21859741]
```

```
[ ]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

      #Make predictions on the test data
     y_pred = lr.predict(X_test)

     # Calculate evaluation metrics
     mae = mean_absolute_error(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     rmse = np.sqrt(mse)
     r_squared = r2_score(y_test, y_pred)

     # Print the evaluation metrics
     print("Mean Absolute Error (MAE):", mae)
     print("Mean Squared Error (MSE):", mse)
     print("Root Mean Squared Error (RMSE):", rmse)
     print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 102802.93500269356
Mean Squared Error (MSE): 36781260236.32816
Root Mean Squared Error (RMSE): 191784.41082717897
R-squared (R²): 0.7266003710319172
```

Xgboost

```
[ ]: import xgboost as xgb
```

```
[ ]: xgb_regressor = xgb.XGBRegressor(
         n_estimators=100,  # Number of boosting rounds
```

```
    learning_rate=0.1,  # Step size shrinkage used in boosting
    max_depth=3,  # Maximum depth of trees
    random_state=0  # Seed for reproducibility
)
xgb_regressor.fit(X_train, y_train)
```

[ ]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.1, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=3, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=100, n_jobs=None,
             num_parallel_tree=None, random_state=0, …)
```

[ ]:
```
y_pred = xgb_regressor.predict(X_test)
y_pred
```

[ ]:
```
array([393484.  , 692280.4 , 725170.2 , …, 572404.7 , 285791.78,
       765646.4 ], dtype=float32)
```

[ ]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

 #Make predictions on the test data
y_pred = xgb_regressor.predict(X_test)

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)

# Print the evaluation metrics
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 14302.099930888053
Mean Squared Error (MSE): 3122306613.658067
Root Mean Squared Error (RMSE): 55877.603864679695
R-squared (R²): 0.9767915111060935
```

Ridge

```
[ ]: from sklearn.linear_model import Ridge
     from sklearn.linear_model import Lasso
     from sklearn.preprocessing import StandardScaler


     scaler = StandardScaler()

     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
```

```
[ ]: alpha = 1.0
     ridge_reg = Ridge(alpha=alpha)
     ridge_reg.fit(X_train_scaled, y_train)
```

```
[ ]: Ridge()
```

```
[ ]: y_pred = ridge_reg.predict(X_test_scaled)
     y_pred
```

```
[ ]: array([328376.55319653, 800118.83230763, 809865.8485878 , …,
            693742.11498351, 306460.47189335, 824197.27118969])
```

```
[ ]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

      #Make predictions on the test data
     y_pred = ridge_reg.predict(X_test)

     # Calculate evaluation metrics
     mae = mean_absolute_error(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     rmse = np.sqrt(mse)
     r_squared = r2_score(y_test, y_pred)

     # Print the evaluation metrics
     print("Mean Absolute Error (MAE):", mae)
     print("Mean Squared Error (MSE):", mse)
     print("Root Mean Squared Error (RMSE):", rmse)
     print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 986891385815659.4
Mean Squared Error (MSE): 9.739546073981897e+29
Root Mean Squared Error (RMSE): 986891385816184.8
R-squared (R²): -7.23952432797344e+18
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:458: UserWarning: X has
feature names, but Ridge was fitted without feature names
  warnings.warn(
```

Lasso

[ ]:
```
alpha = 1.0
lasso_reg = Lasso(alpha=alpha)
lasso_reg.fit(X_train_scaled, y_train)
```

```
/usr/local/lib/python3.10/dist-
packages/sklearn/linear_model/_coordinate_descent.py:628: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 3.586e+13, tolerance: 1.381e+11
  model = cd_fast.enet_coordinate_descent(
```

[ ]:
```
Lasso()
```

[ ]:
```
y_pred = lasso_reg.predict(X_test_scaled)
y_pred
```

[ ]:
```
array([328385.39336167, 800149.05015516, 809901.88280138, …,
       693736.92534209, 306438.37718305, 824212.42037513])
```

[ ]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

 #Make predictions on the test data
y_pred = lasso_reg.predict(X_test)

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)

# Print the evaluation metrics
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 987100855879531.8
Mean Squared Error (MSE): 9.743680996791694e+29
Root Mean Squared Error (RMSE): 987100855880071.4
R-squared (R²): -7.242597866929817e+18
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:458: UserWarning: X has
feature names, but Lasso was fitted without feature names
  warnings.warn(
```

Polynomial regression

[ ]:
```python
from sklearn.preprocessing import PolynomialFeatures, StandardScaler
```

[ ]:
```python
# Create polynomial features with degree 2
poly = PolynomialFeatures(degree=2)
X_train_poly = poly.fit_transform(X_train)
X_test_poly = poly.transform(X_test)

# Train the model
model = LinearRegression()
model.fit(X_train_poly, y_train)

# Make predictions
y_pred = model.predict(X_test_poly)
```

[ ]:
```python
#Make predictions on the test data
y_pred = model.predict(X_test_poly) # Use X_test_poly which has the polynomial␣
↪features

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)

# Print the evaluation metrics
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 67640.14882580939
Mean Squared Error (MSE): 17098315850.300467
Root Mean Squared Error (RMSE): 130760.52864033729
R-squared (R²): 0.8729061163370866
```

[ ]:
```python
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score




# Create a Decision Tree Regressor
tree_regressor = DecisionTreeRegressor()

# Train the model
tree_regressor.fit(X_train, y_train)

# Make predictions on the test set
```

```
y_pred = tree_regressor.predict(X_test)

# Calculate and print metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R2): {r2}")
```

```
Mean Absolute Error (MAE): 18376.762653898768
Mean Squared Error (MSE): 5817462585.210898
R-squared (R2): 0.9567580854458743
```

SVR

[ ]:
```
from sklearn.svm import SVR
# Create an SVR model with your choice of kernel (e.g., 'linear', 'rbf', 'poly')
svr = SVR(kernel='rbf')
```

[ ]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[ ]:
```
svr.fit(X_train_scaled, y_train)
```

[ ]:
```
SVR()
```

[ ]:
```
y_pred = svr.predict(X_test_scaled)
y_pred
```

[ ]:
```
array([449636.97908902, 450244.05035739, 450828.32532136, …,
       450544.73966543, 449601.23508011, 450953.74983207])
```

[ ]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_score

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
```

```python
# Calculate R-squared (R²)
r_squared = r2_score(y_test, y_pred)

print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error: 217758.92196069882
Mean Squared Error: 141981884809.91953
Root Mean Squared Error: 376804.8365001696
R-squared (R²): -0.05536880405423039
```

KNN

[ ]:
```python
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
X_train.shape, X_test.shape
```

[ ]:
```
((11695, 22), (2924, 22))
```

[ ]:
```python
# Create a KNNR model with the desired number of neighbors (k)
knn_regressor = KNeighborsRegressor(n_neighbors=5)
```

[ ]:
```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[ ]:
```python
knn_regressor.fit(X_train_scaled, y_train)
```

[ ]:
```
KNeighborsRegressor()
```

[ ]:
```python
# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# Calculate R-squared (R²)
r_squared = r2_score(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
```

```
mse = mean_squared_error(y_test, y_pred)


print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error: 78130.4021887825
Mean Squared Error: 28591586319.31357
Root Mean Squared Error: 169090.46785467703
R-squared (R²): 0.8068084780083291
```

Random Forest

[ ]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
X_train.shape, X_test.shape
```

[ ]:
```
((11695, 22), (2924, 22))
```

[ ]:
```
# Create a Random Forest Regressor model with the desired number of trees␣
↪(n_estimators)
random_forest = RandomForestRegressor(n_estimators=100)
random_forest.fit(X_train, y_train)
```

[ ]:
```
RandomForestRegressor()
```

[ ]:
```
y_pred = random_forest.predict(X_test)
y_pred
```

[ ]:
```
array([ 535768.32,  552728.31,  393552.62, …,  693271.5 ,  442066.  ,
       1922200.  ])
```

[ ]:
```
# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate R-squared (R²)
r_squared = r2_score(y_test, y_pred)
```

```
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error: 15774.420530095758
Mean Squared Error: 4827167718.002883
Root Mean Squared Error: 69477.82177071244
R-squared (R²): 0.9673831361458917
```

Gradient Boosting

```
[ ]: from sklearn.ensemble import GradientBoostingRegressor
```

```
[ ]: # Create a Gradient Boosting Regressor model with desired hyperparameters
gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,␣
 ↪max_depth=3, random_state=0)
gbr.fit(X_train, y_train)
```

```
[ ]: GradientBoostingRegressor(random_state=0)
```

```
[ ]: y_pred = gbr.predict(X_test)
y_pred
```

```
[ ]: array([ 536698.58956357,  552471.50909102,  394730.73051084, …,
        693567.68617714,  443622.03399905, 1989059.51955696])
```

```
[ ]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate R-squared (R²)
r_squared = r2_score(y_test, y_pred)

print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
```

```
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error: 15842.340413691856
Mean Squared Error: 4092564435.616683
Root Mean Squared Error: 63973.15402273584
R-squared (R²): 0.9723468035898488
```

AdaBoost

[ ]:
```
from sklearn.ensemble import AdaBoostRegressor
```

[ ]:
```
# Create and train an AdaBoost regressor
model = AdaBoostRegressor(n_estimators=100, random_state=42,learning_rate=0.1)
model.fit(X_train, y_train)
```

[ ]:
```
AdaBoostRegressor(learning_rate=0.1, n_estimators=100, random_state=42)
```

[ ]:
```
# Make predictions on the testing set
y_pred = model.predict(X_test)
```

[ ]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)

# Print the evaluation metrics
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R²):", r_squared)
```

```
Mean Absolute Error (MAE): 63296.29838565224
Mean Squared Error (MSE): 10148598374.359457
Root Mean Squared Error (RMSE): 100740.25200663068
R-squared (R²): 0.9314265691966764
```

ANN

[ ]:
```
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
 ↪random_state=42)
X_train.shape, X_test.shape
```

```python
# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Build the neural network model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='linear',input_shape=(X_train.
 ↪shape[1],)),
    tf.keras.layers.Dense(32, activation='linear'),
    tf.keras.layers.Dense(1, activation='linear')
])

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
# Train the model
history = model.fit(X_train, y_train, epochs=500, batch_size=32,␣
 ↪validation_data=(X_test, y_test))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)

Epoch 1/500
343/343              3s 5ms/step -
loss: 413646684160.0000 - val_loss: 444322447360.0000
Epoch 2/500
343/343              2s 4ms/step -
loss: 396074221568.0000 - val_loss: 430079541248.0000
Epoch 3/500
343/343              2s 4ms/step -
loss: 403503874048.0000 - val_loss: 400431841280.0000
Epoch 4/500
343/343              3s 5ms/step -
loss: 349063348224.0000 - val_loss: 360441315328.0000
Epoch 5/500
343/343              3s 7ms/step -
loss: 331318493184.0000 - val_loss: 321497038848.0000
Epoch 6/500
343/343              2s 6ms/step -
loss: 288426917888.0000 - val_loss: 290209595392.0000
Epoch 7/500
343/343              2s 5ms/step -
loss: 264384249856.0000 - val_loss: 265734356992.0000
Epoch 8/500
```

```
343/343 3s 5ms/step -
loss: 243794067456.0000 - val_loss: 244681801728.0000
Epoch 9/500
343/343 2s 4ms/step -
loss: 230141640704.0000 - val_loss: 223407177728.0000
Epoch 10/500
343/343 2s 2ms/step -
loss: 202920394752.0000 - val_loss: 200805957632.0000
Epoch 11/500
343/343 2s 4ms/step -
loss: 177932746752.0000 - val_loss: 176522297344.0000
Epoch 12/500
343/343 1s 4ms/step -
loss: 153638715392.0000 - val_loss: 151207346176.0000
Epoch 13/500
343/343 2s 2ms/step -
loss: 130119442432.0000 - val_loss: 125576822784.0000
Epoch 14/500
343/343 1s 3ms/step -
loss: 100092141568.0000 - val_loss: 101011456000.0000
Epoch 15/500
343/343 1s 3ms/step -
loss: 84737384448.0000 - val_loss: 80240377856.0000
Epoch 16/500
343/343 1s 2ms/step -
loss: 65606868992.0000 - val_loss: 64335790080.0000
Epoch 17/500
343/343 1s 2ms/step -
loss: 50455252992.0000 - val_loss: 53580431360.0000
Epoch 18/500
343/343 1s 2ms/step -
loss: 39516000256.0000 - val_loss: 47536070656.0000
Epoch 19/500
343/343 1s 2ms/step -
loss: 35442987008.0000 - val_loss: 44824530944.0000
Epoch 20/500
343/343 1s 3ms/step -
loss: 33852788736.0000 - val_loss: 43629166592.0000
Epoch 21/500
343/343 1s 4ms/step -
loss: 34439188480.0000 - val_loss: 43217690624.0000
Epoch 22/500
343/343 2s 2ms/step -
loss: 29647394816.0000 - val_loss: 42961534976.0000
Epoch 23/500
343/343 1s 2ms/step -
loss: 29796347904.0000 - val_loss: 42883653632.0000
Epoch 24/500
```

```
343/343             1s 2ms/step -
loss: 31290118144.0000 - val_loss: 42966167552.0000
Epoch 25/500
343/343             1s 2ms/step -
loss: 28248590336.0000 - val_loss: 42847223808.0000
Epoch 26/500
343/343             1s 2ms/step -
loss: 33799991296.0000 - val_loss: 42814525440.0000
Epoch 27/500
343/343             1s 2ms/step -
loss: 34734252032.0000 - val_loss: 43033395200.0000
Epoch 28/500
343/343             1s 2ms/step -
loss: 34177241088.0000 - val_loss: 42910904320.0000
Epoch 29/500
343/343             1s 2ms/step -
loss: 33934170112.0000 - val_loss: 42837299200.0000
Epoch 30/500
343/343             1s 2ms/step -
loss: 28556978176.0000 - val_loss: 42774302720.0000
Epoch 31/500
343/343             2s 4ms/step -
loss: 28549113856.0000 - val_loss: 42713387008.0000
Epoch 32/500
343/343             1s 4ms/step -
loss: 34004195328.0000 - val_loss: 42722189312.0000
Epoch 33/500
343/343             1s 3ms/step -
loss: 38499508224.0000 - val_loss: 42990497792.0000
Epoch 34/500
343/343             1s 2ms/step -
loss: 31031134208.0000 - val_loss: 42889113600.0000
Epoch 35/500
343/343             1s 2ms/step -
loss: 31858966528.0000 - val_loss: 42986770432.0000
Epoch 36/500
343/343             1s 2ms/step -
loss: 29485121536.0000 - val_loss: 42884546560.0000
Epoch 37/500
343/343             1s 3ms/step -
loss: 31355949056.0000 - val_loss: 42887458816.0000
Epoch 38/500
343/343             1s 2ms/step -
loss: 26831446016.0000 - val_loss: 42878324736.0000
Epoch 39/500
343/343             1s 2ms/step -
loss: 36124397568.0000 - val_loss: 42985189376.0000
Epoch 40/500
```

```
343/343              1s 2ms/step -
loss: 31201155072.0000 - val_loss: 42856501248.0000
Epoch 41/500
343/343              2s 3ms/step -
loss: 34346856448.0000 - val_loss: 43008987136.0000
Epoch 42/500
343/343              1s 4ms/step -
loss: 29449496576.0000 - val_loss: 42941530112.0000
Epoch 43/500
343/343              2s 2ms/step -
loss: 37918564352.0000 - val_loss: 43086082048.0000
Epoch 44/500
343/343              1s 3ms/step -
loss: 31662532608.0000 - val_loss: 42971672576.0000
Epoch 45/500
343/343              1s 2ms/step -
loss: 30084220928.0000 - val_loss: 42906185728.0000
Epoch 46/500
343/343              1s 2ms/step -
loss: 30915022848.0000 - val_loss: 42929963008.0000
Epoch 47/500
343/343              1s 2ms/step -
loss: 31613923328.0000 - val_loss: 42981445632.0000
Epoch 48/500
343/343              1s 2ms/step -
loss: 32783323136.0000 - val_loss: 42840920064.0000
Epoch 49/500
343/343              1s 2ms/step -
loss: 32934979584.0000 - val_loss: 42970980352.0000
Epoch 50/500
343/343              1s 2ms/step -
loss: 28553674752.0000 - val_loss: 42912702464.0000
Epoch 51/500
343/343              1s 2ms/step -
loss: 28664080384.0000 - val_loss: 42872152064.0000
Epoch 52/500
343/343              1s 3ms/step -
loss: 27913326592.0000 - val_loss: 42826473472.0000
Epoch 53/500
343/343              2s 4ms/step -
loss: 34583375872.0000 - val_loss: 42818584576.0000
Epoch 54/500
343/343              2s 2ms/step -
loss: 29926199296.0000 - val_loss: 42930221056.0000
Epoch 55/500
343/343              1s 2ms/step -
loss: 33118740480.0000 - val_loss: 42848198656.0000
Epoch 56/500
```

```
343/343            1s 2ms/step -
loss: 32196397056.0000 - val_loss: 42928521216.0000
Epoch 57/500
343/343            1s 2ms/step -
loss: 26919929856.0000 - val_loss: 42760830976.0000
Epoch 58/500
343/343            1s 2ms/step -
loss: 29591099392.0000 - val_loss: 42882498560.0000
Epoch 59/500
343/343            1s 2ms/step -
loss: 32965337088.0000 - val_loss: 42899009536.0000
Epoch 60/500
343/343            1s 2ms/step -
loss: 33642326016.0000 - val_loss: 42896347136.0000
Epoch 61/500
343/343            1s 2ms/step -
loss: 30176966656.0000 - val_loss: 42939129856.0000
Epoch 62/500
343/343            1s 3ms/step -
loss: 31341025280.0000 - val_loss: 42891083776.0000
Epoch 63/500
343/343            2s 4ms/step -
loss: 33099882496.0000 - val_loss: 42912800768.0000
Epoch 64/500
343/343            2s 3ms/step -
loss: 34240849920.0000 - val_loss: 42890416128.0000
Epoch 65/500
343/343            1s 2ms/step -
loss: 31998459904.0000 - val_loss: 43019411456.0000
Epoch 66/500
343/343            1s 2ms/step -
loss: 32278558720.0000 - val_loss: 42947698688.0000
Epoch 67/500
343/343            1s 2ms/step -
loss: 32732391424.0000 - val_loss: 42997293056.0000
Epoch 68/500
343/343            1s 2ms/step -
loss: 29811021824.0000 - val_loss: 42996572160.0000
Epoch 69/500
343/343            1s 2ms/step -
loss: 36279812096.0000 - val_loss: 42974072832.0000
Epoch 70/500
343/343            1s 2ms/step -
loss: 31741417472.0000 - val_loss: 42974580736.0000
Epoch 71/500
343/343            1s 2ms/step -
loss: 37247995904.0000 - val_loss: 42926784512.0000
Epoch 72/500
```

```
343/343              2s 3ms/step -
loss: 29918922752.0000 - val_loss: 42960121856.0000
Epoch 73/500
343/343              1s 4ms/step -
loss: 27882749952.0000 - val_loss: 42821185536.0000
Epoch 74/500
343/343              2s 2ms/step -
loss: 31577481216.0000 - val_loss: 42920808448.0000
Epoch 75/500
343/343              1s 2ms/step -
loss: 30984333312.0000 - val_loss: 42914988032.0000
Epoch 76/500
343/343              1s 2ms/step -
loss: 34502987776.0000 - val_loss: 42965508096.0000
Epoch 77/500
343/343              1s 2ms/step -
loss: 34141724672.0000 - val_loss: 42900688896.0000
Epoch 78/500
343/343              1s 2ms/step -
loss: 30337241088.0000 - val_loss: 42900733952.0000
Epoch 79/500
343/343              1s 2ms/step -
loss: 34844786688.0000 - val_loss: 42971017216.0000
Epoch 80/500
343/343              1s 2ms/step -
loss: 31509637120.0000 - val_loss: 42877820928.0000
Epoch 81/500
343/343              2s 3ms/step -
loss: 30652774400.0000 - val_loss: 43019014144.0000
Epoch 82/500
343/343              1s 4ms/step -
loss: 29217103872.0000 - val_loss: 42950455296.0000
Epoch 83/500
343/343              1s 3ms/step -
loss: 29080340480.0000 - val_loss: 42854940672.0000
Epoch 84/500
343/343              1s 3ms/step -
loss: 34649612288.0000 - val_loss: 43056029696.0000
Epoch 85/500
343/343              1s 2ms/step -
loss: 29101004800.0000 - val_loss: 42829656064.0000
Epoch 86/500
343/343              1s 2ms/step -
loss: 32631040000.0000 - val_loss: 42921570304.0000
Epoch 87/500
343/343              1s 2ms/step -
loss: 43210027008.0000 - val_loss: 43007676416.0000
Epoch 88/500
```

```
343/343             1s 2ms/step -
loss: 29507170304.0000 - val_loss: 42888839168.0000
Epoch 89/500
343/343             1s 2ms/step -
loss: 37343997952.0000 - val_loss: 42934710272.0000
Epoch 90/500
343/343             1s 2ms/step -
loss: 31013097472.0000 - val_loss: 42965979136.0000
Epoch 91/500
343/343             1s 2ms/step -
loss: 33854023680.0000 - val_loss: 43126276096.0000
Epoch 92/500
343/343             1s 2ms/step -
loss: 29434005504.0000 - val_loss: 42887704576.0000
Epoch 93/500
343/343             1s 3ms/step -
loss: 28246276096.0000 - val_loss: 42828525568.0000
Epoch 94/500
343/343             1s 3ms/step -
loss: 32975517696.0000 - val_loss: 42795384832.0000
Epoch 95/500
343/343             1s 4ms/step -
loss: 30216970240.0000 - val_loss: 42828443648.0000
Epoch 96/500
343/343             2s 2ms/step -
loss: 29889890304.0000 - val_loss: 42888429568.0000
Epoch 97/500
343/343             1s 2ms/step -
loss: 28423792640.0000 - val_loss: 42822221824.0000
Epoch 98/500
343/343             1s 2ms/step -
loss: 31201378304.0000 - val_loss: 42956570624.0000
Epoch 99/500
343/343             2s 4ms/step -
loss: 29956216832.0000 - val_loss: 42917404672.0000
Epoch 100/500
343/343             1s 2ms/step -
loss: 36155465728.0000 - val_loss: 43011493888.0000
Epoch 101/500
343/343             1s 2ms/step -
loss: 32087351296.0000 - val_loss: 42837245952.0000
Epoch 102/500
343/343             1s 2ms/step -
loss: 33881845760.0000 - val_loss: 42967982080.0000
Epoch 103/500
343/343             1s 2ms/step -
loss: 32449667072.0000 - val_loss: 42963771392.0000
Epoch 104/500
```

```
343/343             2s 4ms/step -
loss: 28173613056.0000 - val_loss: 42849402880.0000
Epoch 105/500
343/343             1s 4ms/step -
loss: 33515311104.0000 - val_loss: 42939105280.0000
Epoch 106/500
343/343             2s 4ms/step -
loss: 26968948736.0000 - val_loss: 42805981184.0000
Epoch 107/500
343/343             2s 2ms/step -
loss: 33018449920.0000 - val_loss: 42813046784.0000
Epoch 108/500
343/343             1s 2ms/step -
loss: 31267026944.0000 - val_loss: 42822041600.0000
Epoch 109/500
343/343             1s 2ms/step -
loss: 31352616960.0000 - val_loss: 42820427776.0000
Epoch 110/500
343/343             1s 2ms/step -
loss: 32794478592.0000 - val_loss: 42878066688.0000
Epoch 111/500
343/343             1s 2ms/step -
loss: 29338761216.0000 - val_loss: 42782531584.0000
Epoch 112/500
343/343             1s 2ms/step -
loss: 34065092608.0000 - val_loss: 42898206720.0000
Epoch 113/500
343/343             2s 3ms/step -
loss: 29005406208.0000 - val_loss: 42887430144.0000
Epoch 114/500
343/343             1s 3ms/step -
loss: 33740623872.0000 - val_loss: 42948685824.0000
Epoch 115/500
343/343             1s 4ms/step -
loss: 32382320640.0000 - val_loss: 42932334592.0000
Epoch 116/500
343/343             2s 2ms/step -
loss: 30846613504.0000 - val_loss: 42896941056.0000
Epoch 117/500
343/343             1s 2ms/step -
loss: 36332793856.0000 - val_loss: 43001004032.0000
Epoch 118/500
343/343             1s 2ms/step -
loss: 31167537152.0000 - val_loss: 42808791040.0000
Epoch 119/500
343/343             1s 3ms/step -
loss: 28888293376.0000 - val_loss: 42828431360.0000
Epoch 120/500
```

```
343/343            1s 2ms/step -
loss: 33709090816.0000 - val_loss: 42966220800.0000
Epoch 121/500
343/343            1s 2ms/step -
loss: 34293452800.0000 - val_loss: 42946150400.0000
Epoch 122/500
343/343            1s 2ms/step -
loss: 33522012160.0000 - val_loss: 43025244160.0000
Epoch 123/500
343/343            1s 2ms/step -
loss: 28024635392.0000 - val_loss: 42810253312.0000
Epoch 124/500
343/343            3s 7ms/step -
loss: 28159918080.0000 - val_loss: 42846781440.0000
Epoch 125/500
343/343            1s 3ms/step -
loss: 29421441024.0000 - val_loss: 42865250304.0000
Epoch 126/500
343/343            1s 2ms/step -
loss: 30680100864.0000 - val_loss: 42956439552.0000
Epoch 127/500
343/343            1s 2ms/step -
loss: 36668604416.0000 - val_loss: 42968604672.0000
Epoch 128/500
343/343            1s 2ms/step -
loss: 32345708544.0000 - val_loss: 42961489920.0000
Epoch 129/500
343/343            1s 2ms/step -
loss: 30976874496.0000 - val_loss: 42874634240.0000
Epoch 130/500
343/343            1s 2ms/step -
loss: 31620784128.0000 - val_loss: 42833113088.0000
Epoch 131/500
343/343            1s 2ms/step -
loss: 31142723584.0000 - val_loss: 42907648000.0000
Epoch 132/500
343/343            1s 2ms/step -
loss: 37145948160.0000 - val_loss: 42906820608.0000
Epoch 133/500
343/343            1s 2ms/step -
loss: 28429699072.0000 - val_loss: 42830422016.0000
Epoch 134/500
343/343            2s 4ms/step -
loss: 32293513216.0000 - val_loss: 42969915392.0000
Epoch 135/500
343/343            2s 2ms/step -
loss: 39124987904.0000 - val_loss: 43013857280.0000
Epoch 136/500
```

```
343/343             1s 2ms/step -
loss: 32779425792.0000 - val_loss: 42936123392.0000
Epoch 137/500
343/343             1s 2ms/step -
loss: 32259622912.0000 - val_loss: 42924912640.0000
Epoch 138/500
343/343             1s 2ms/step -
loss: 27825936384.0000 - val_loss: 42925514752.0000
Epoch 139/500
343/343             1s 2ms/step -
loss: 33028790272.0000 - val_loss: 42917675008.0000
Epoch 140/500
343/343             1s 2ms/step -
loss: 35005415424.0000 - val_loss: 42942697472.0000
Epoch 141/500
343/343             1s 2ms/step -
loss: 41259667456.0000 - val_loss: 43110707200.0000
Epoch 142/500
343/343             1s 2ms/step -
loss: 34406764544.0000 - val_loss: 43071176704.0000
Epoch 143/500
343/343             1s 2ms/step -
loss: 31212290048.0000 - val_loss: 42886664192.0000
Epoch 144/500
343/343             1s 3ms/step -
loss: 31241467904.0000 - val_loss: 42844205056.0000
Epoch 145/500
343/343             2s 4ms/step -
loss: 31868250112.0000 - val_loss: 42810011648.0000
Epoch 146/500
343/343             2s 2ms/step -
loss: 31431051264.0000 - val_loss: 42871394304.0000
Epoch 147/500
343/343             1s 2ms/step -
loss: 30839834624.0000 - val_loss: 42847997952.0000
Epoch 148/500
343/343             1s 2ms/step -
loss: 27918075904.0000 - val_loss: 42814754816.0000
Epoch 149/500
343/343             1s 2ms/step -
loss: 32906954752.0000 - val_loss: 42928680960.0000
Epoch 150/500
343/343             1s 2ms/step -
loss: 28438710272.0000 - val_loss: 42807721984.0000
Epoch 151/500
343/343             1s 2ms/step -
loss: 30834044928.0000 - val_loss: 42869297152.0000
Epoch 152/500
```

```
343/343              1s 2ms/step -
loss: 30099644416.0000 - val_loss: 42866827264.0000
Epoch 153/500
343/343              2s 3ms/step -
loss: 31168450560.0000 - val_loss: 42801995776.0000
Epoch 154/500
343/343              2s 4ms/step -
loss: 26798372864.0000 - val_loss: 42833084416.0000
Epoch 155/500
343/343              2s 2ms/step -
loss: 34629214208.0000 - val_loss: 42905583616.0000
Epoch 156/500
343/343              1s 2ms/step -
loss: 38693064704.0000 - val_loss: 42941673472.0000
Epoch 157/500
343/343              1s 2ms/step -
loss: 32196694016.0000 - val_loss: 42885349376.0000
Epoch 158/500
343/343              1s 2ms/step -
loss: 28408186880.0000 - val_loss: 42814668800.0000
Epoch 159/500
343/343              1s 2ms/step -
loss: 35454754816.0000 - val_loss: 42944331776.0000
Epoch 160/500
343/343              1s 2ms/step -
loss: 29425516544.0000 - val_loss: 42836140032.0000
Epoch 161/500
343/343              1s 2ms/step -
loss: 32682450944.0000 - val_loss: 42863558656.0000
Epoch 162/500
343/343              1s 2ms/step -
loss: 29801383936.0000 - val_loss: 42848161792.0000
Epoch 163/500
343/343              2s 4ms/step -
loss: 31149449216.0000 - val_loss: 42881900544.0000
Epoch 164/500
343/343              1s 4ms/step -
loss: 32987547648.0000 - val_loss: 42851876864.0000
Epoch 165/500
343/343              2s 2ms/step -
loss: 30575562752.0000 - val_loss: 42830909440.0000
Epoch 166/500
343/343              1s 2ms/step -
loss: 30087147520.0000 - val_loss: 42820521984.0000
Epoch 167/500
343/343              1s 2ms/step -
loss: 32241649664.0000 - val_loss: 42926301184.0000
Epoch 168/500
```

```
343/343               1s 2ms/step -
loss: 30076116992.0000 - val_loss: 42835513344.0000
Epoch 169/500
343/343               1s 2ms/step -
loss: 31754303488.0000 - val_loss: 42963357696.0000
Epoch 170/500
343/343               1s 2ms/step -
loss: 26592415744.0000 - val_loss: 42840608768.0000
Epoch 171/500
343/343               1s 2ms/step -
loss: 33847773184.0000 - val_loss: 42895413248.0000
Epoch 172/500
343/343               1s 2ms/step -
loss: 44691685376.0000 - val_loss: 43070296064.0000
Epoch 173/500
343/343               2s 4ms/step -
loss: 27121489920.0000 - val_loss: 42940661760.0000
Epoch 174/500
343/343               1s 4ms/step -
loss: 31196698624.0000 - val_loss: 42874757120.0000
Epoch 175/500
343/343               1s 3ms/step -
loss: 28955953152.0000 - val_loss: 42920820736.0000
Epoch 176/500
343/343               1s 3ms/step -
loss: 28247046144.0000 - val_loss: 42813603840.0000
Epoch 177/500
343/343               1s 2ms/step -
loss: 28562548736.0000 - val_loss: 42747191296.0000
Epoch 178/500
343/343               1s 2ms/step -
loss: 32298332160.0000 - val_loss: 42915065856.0000
Epoch 179/500
343/343               1s 2ms/step -
loss: 30698139648.0000 - val_loss: 42861867008.0000
Epoch 180/500
343/343               1s 2ms/step -
loss: 31522822144.0000 - val_loss: 42832470016.0000
Epoch 181/500
343/343               1s 2ms/step -
loss: 34043930624.0000 - val_loss: 43004317696.0000
Epoch 182/500
343/343               1s 2ms/step -
loss: 30914146304.0000 - val_loss: 42905882624.0000
Epoch 183/500
343/343               1s 2ms/step -
loss: 30228205568.0000 - val_loss: 42973847552.0000
Epoch 184/500
```

```
343/343              2s 3ms/step -
loss: 29263736832.0000 - val_loss: 42951622656.0000
Epoch 185/500
343/343              1s 4ms/step -
loss: 31660857344.0000 - val_loss: 43094802432.0000
Epoch 186/500
343/343              2s 2ms/step -
loss: 30384549888.0000 - val_loss: 42948333568.0000
Epoch 187/500
343/343              1s 2ms/step -
loss: 28862093312.0000 - val_loss: 42826170368.0000
Epoch 188/500
343/343              1s 2ms/step -
loss: 30693545984.0000 - val_loss: 42858954752.0000
Epoch 189/500
343/343              1s 2ms/step -
loss: 28138143744.0000 - val_loss: 42857713664.0000
Epoch 190/500
343/343              1s 2ms/step -
loss: 35660537856.0000 - val_loss: 42899259392.0000
Epoch 191/500
343/343              1s 2ms/step -
loss: 35655643136.0000 - val_loss: 42963841024.0000
Epoch 192/500
343/343              1s 2ms/step -
loss: 37721903104.0000 - val_loss: 42973765632.0000
Epoch 193/500
343/343              1s 2ms/step -
loss: 32362106880.0000 - val_loss: 42920275968.0000
Epoch 194/500
343/343              2s 3ms/step -
loss: 32417146880.0000 - val_loss: 42931048448.0000
Epoch 195/500
343/343              1s 4ms/step -
loss: 37399289856.0000 - val_loss: 42921701376.0000
Epoch 196/500
343/343              2s 2ms/step -
loss: 27300814848.0000 - val_loss: 42844372992.0000
Epoch 197/500
343/343              1s 2ms/step -
loss: 34947403776.0000 - val_loss: 42846998528.0000
Epoch 198/500
343/343              1s 2ms/step -
loss: 31305627648.0000 - val_loss: 42915864576.0000
Epoch 199/500
343/343              1s 2ms/step -
loss: 27384238080.0000 - val_loss: 42781298688.0000
Epoch 200/500
```

```
343/343              1s 2ms/step -
loss: 32481370112.0000 - val_loss: 42930147328.0000
Epoch 201/500
343/343              1s 2ms/step -
loss: 29297854464.0000 - val_loss: 42867847168.0000
Epoch 202/500
343/343              1s 2ms/step -
loss: 33275174912.0000 - val_loss: 42951487488.0000
Epoch 203/500
343/343              1s 2ms/step -
loss: 31049771008.0000 - val_loss: 42907443200.0000
Epoch 204/500
343/343              1s 2ms/step -
loss: 28824989696.0000 - val_loss: 42903191552.0000
Epoch 205/500
343/343              2s 3ms/step -
loss: 28186243072.0000 - val_loss: 42807566336.0000
Epoch 206/500
343/343              1s 4ms/step -
loss: 34196242432.0000 - val_loss: 42961063936.0000
Epoch 207/500
343/343              1s 4ms/step -
loss: 33450459136.0000 - val_loss: 43070246912.0000
Epoch 208/500
343/343              2s 2ms/step -
loss: 29318989824.0000 - val_loss: 42798047232.0000
Epoch 209/500
343/343              1s 2ms/step -
loss: 30319769600.0000 - val_loss: 42906636288.0000
Epoch 210/500
343/343              1s 2ms/step -
loss: 38063820800.0000 - val_loss: 43166736384.0000
Epoch 211/500
343/343              1s 2ms/step -
loss: 28948830208.0000 - val_loss: 42908045312.0000
Epoch 212/500
343/343              1s 2ms/step -
loss: 31677308928.0000 - val_loss: 42847342592.0000
Epoch 213/500
343/343              1s 2ms/step -
loss: 37077348352.0000 - val_loss: 42859368448.0000
Epoch 214/500
343/343              1s 2ms/step -
loss: 35968282624.0000 - val_loss: 43014316032.0000
Epoch 215/500
343/343              2s 3ms/step -
loss: 32832651264.0000 - val_loss: 43062398976.0000
Epoch 216/500
```

```
343/343              2s 4ms/step -
loss: 30993326080.0000 - val_loss: 42924568576.0000
Epoch 217/500
343/343              2s 2ms/step -
loss: 28151668736.0000 - val_loss: 42835124224.0000
Epoch 218/500
343/343              1s 2ms/step -
loss: 31506198528.0000 - val_loss: 42884816896.0000
Epoch 219/500
343/343              1s 2ms/step -
loss: 32296495104.0000 - val_loss: 42909085696.0000
Epoch 220/500
343/343              1s 2ms/step -
loss: 27381016576.0000 - val_loss: 42798587904.0000
Epoch 221/500
343/343              2s 6ms/step -
loss: 27398328320.0000 - val_loss: 42762612736.0000
Epoch 222/500
343/343              2s 5ms/step -
loss: 34009841664.0000 - val_loss: 42927796224.0000
Epoch 223/500
343/343              2s 7ms/step -
loss: 28583368704.0000 - val_loss: 42807861248.0000
Epoch 224/500
343/343              2s 7ms/step -
loss: 31533164544.0000 - val_loss: 42963906560.0000
Epoch 225/500
343/343              3s 7ms/step -
loss: 37724274688.0000 - val_loss: 42970484736.0000
Epoch 226/500
343/343              2s 5ms/step -
loss: 34948194304.0000 - val_loss: 42975920128.0000
Epoch 227/500
343/343              1s 2ms/step -
loss: 28715079680.0000 - val_loss: 42877718528.0000
Epoch 228/500
343/343              1s 2ms/step -
loss: 27438374912.0000 - val_loss: 42808160256.0000
Epoch 229/500
343/343              1s 2ms/step -
loss: 29712615424.0000 - val_loss: 42831634432.0000
Epoch 230/500
343/343              1s 2ms/step -
loss: 33449523200.0000 - val_loss: 42904580096.0000
Epoch 231/500
343/343              1s 2ms/step -
loss: 27657678848.0000 - val_loss: 42809679872.0000
Epoch 232/500
```

```
343/343              1s 2ms/step -
loss: 35860123648.0000 - val_loss: 42953547776.0000
Epoch 233/500
343/343              2s 3ms/step -
loss: 32873906176.0000 - val_loss: 42893594624.0000
Epoch 234/500
343/343              1s 3ms/step -
loss: 33084139520.0000 - val_loss: 42914512896.0000
Epoch 235/500
343/343              1s 4ms/step -
loss: 31904530432.0000 - val_loss: 42896490496.0000
Epoch 236/500
343/343              1s 3ms/step -
loss: 31399968768.0000 - val_loss: 42826084352.0000
Epoch 237/500
343/343              1s 2ms/step -
loss: 37447421952.0000 - val_loss: 42952146944.0000
Epoch 238/500
343/343              1s 2ms/step -
loss: 32538353664.0000 - val_loss: 42890256384.0000
Epoch 239/500
343/343              1s 2ms/step -
loss: 29317074944.0000 - val_loss: 42845773824.0000
Epoch 240/500
343/343              1s 2ms/step -
loss: 27511814144.0000 - val_loss: 42772832256.0000
Epoch 241/500
343/343              1s 2ms/step -
loss: 30359046144.0000 - val_loss: 42798825472.0000
Epoch 242/500
343/343              1s 2ms/step -
loss: 28568739840.0000 - val_loss: 42823737344.0000
Epoch 243/500
343/343              1s 2ms/step -
loss: 31282831360.0000 - val_loss: 42872545280.0000
Epoch 244/500
343/343              1s 2ms/step -
loss: 35231113216.0000 - val_loss: 42887917568.0000
Epoch 245/500
343/343              1s 2ms/step -
loss: 31965593600.0000 - val_loss: 42978934784.0000
Epoch 246/500
343/343              1s 4ms/step -
loss: 35184234496.0000 - val_loss: 42992058368.0000
Epoch 247/500
343/343              1s 4ms/step -
loss: 36468695040.0000 - val_loss: 42997706752.0000
Epoch 248/500
```

```
343/343              2s 2ms/step -
loss: 35134423040.0000 - val_loss: 43064246272.0000
Epoch 249/500
343/343              1s 2ms/step -
loss: 34407976960.0000 - val_loss: 43036389376.0000
Epoch 250/500
343/343              1s 2ms/step -
loss: 36342472704.0000 - val_loss: 42974265344.0000
Epoch 251/500
343/343              1s 2ms/step -
loss: 28485015552.0000 - val_loss: 42845999104.0000
Epoch 252/500
343/343              1s 2ms/step -
loss: 30914088960.0000 - val_loss: 42937262080.0000
Epoch 253/500
343/343              1s 2ms/step -
loss: 30562484224.0000 - val_loss: 42893156352.0000
Epoch 254/500
343/343              1s 2ms/step -
loss: 29732915200.0000 - val_loss: 42825428992.0000
Epoch 255/500
343/343              1s 2ms/step -
loss: 28040163328.0000 - val_loss: 42788978688.0000
Epoch 256/500
343/343              2s 3ms/step -
loss: 32580259840.0000 - val_loss: 42858393600.0000
Epoch 257/500
343/343              1s 4ms/step -
loss: 32529534976.0000 - val_loss: 42899025920.0000
Epoch 258/500
343/343              1s 4ms/step -
loss: 31874187264.0000 - val_loss: 42860711936.0000
Epoch 259/500
343/343              1s 2ms/step -
loss: 27661086720.0000 - val_loss: 42788118528.0000
Epoch 260/500
343/343              1s 2ms/step -
loss: 32725092352.0000 - val_loss: 42824138752.0000
Epoch 261/500
343/343              1s 2ms/step -
loss: 36582395904.0000 - val_loss: 42887475200.0000
Epoch 262/500
343/343              1s 2ms/step -
loss: 30252206080.0000 - val_loss: 42910650368.0000
Epoch 263/500
343/343              1s 2ms/step -
loss: 31471792128.0000 - val_loss: 42867879936.0000
Epoch 264/500
```

```
343/343              1s 2ms/step -
loss: 37700972544.0000 - val_loss: 42961633280.0000
Epoch 265/500
343/343              1s 2ms/step -
loss: 35647119360.0000 - val_loss: 42963533824.0000
Epoch 266/500
343/343              1s 2ms/step -
loss: 30634833920.0000 - val_loss: 42868572160.0000
Epoch 267/500
343/343              1s 2ms/step -
loss: 27073271808.0000 - val_loss: 42831495168.0000
Epoch 268/500
343/343              1s 4ms/step -
loss: 29812330496.0000 - val_loss: 42845200384.0000
Epoch 269/500
343/343              1s 3ms/step -
loss: 33847261184.0000 - val_loss: 42921877504.0000
Epoch 270/500
343/343              1s 4ms/step -
loss: 30653863936.0000 - val_loss: 42895532032.0000
Epoch 271/500
343/343              2s 2ms/step -
loss: 30764279808.0000 - val_loss: 42925105152.0000
Epoch 272/500
343/343              1s 2ms/step -
loss: 28730861568.0000 - val_loss: 42827558912.0000
Epoch 273/500
343/343              1s 2ms/step -
loss: 35181166592.0000 - val_loss: 43005026304.0000
Epoch 274/500
343/343              1s 2ms/step -
loss: 31015555072.0000 - val_loss: 43026984960.0000
Epoch 275/500
343/343              1s 2ms/step -
loss: 29657266176.0000 - val_loss: 42868756480.0000
Epoch 276/500
343/343              1s 2ms/step -
loss: 31617056768.0000 - val_loss: 42914709504.0000
Epoch 277/500
343/343              1s 2ms/step -
loss: 34281795584.0000 - val_loss: 42945761280.0000
Epoch 278/500
343/343              1s 2ms/step -
loss: 33461770240.0000 - val_loss: 43065917440.0000
Epoch 279/500
343/343              1s 2ms/step -
loss: 29982562304.0000 - val_loss: 42869473280.0000
Epoch 280/500
```

```
343/343          2s 3ms/step -
loss: 31980273664.0000 - val_loss: 42956587008.0000
Epoch 281/500
343/343          1s 4ms/step -
loss: 31030654976.0000 - val_loss: 42921029632.0000
Epoch 282/500
343/343          2s 2ms/step -
loss: 29919463424.0000 - val_loss: 42887725056.0000
Epoch 283/500
343/343          1s 2ms/step -
loss: 29967413248.0000 - val_loss: 42851254272.0000
Epoch 284/500
343/343          1s 2ms/step -
loss: 32530982912.0000 - val_loss: 42936131584.0000
Epoch 285/500
343/343          2s 6ms/step -
loss: 31992121344.0000 - val_loss: 43004710912.0000
Epoch 286/500
343/343          1s 3ms/step -
loss: 31489146880.0000 - val_loss: 42935353344.0000
Epoch 287/500
343/343          1s 3ms/step -
loss: 32413929472.0000 - val_loss: 42948542464.0000
Epoch 288/500
343/343          1s 3ms/step -
loss: 28391434240.0000 - val_loss: 42735132672.0000
Epoch 289/500
343/343          2s 4ms/step -
loss: 32110184448.0000 - val_loss: 42869850112.0000
Epoch 290/500
343/343          2s 4ms/step -
loss: 32502038528.0000 - val_loss: 42841968640.0000
Epoch 291/500
343/343          2s 2ms/step -
loss: 36845920256.0000 - val_loss: 43026116608.0000
Epoch 292/500
343/343          1s 2ms/step -
loss: 31756392448.0000 - val_loss: 42993045504.0000
Epoch 293/500
343/343          1s 2ms/step -
loss: 32541724672.0000 - val_loss: 43058200576.0000
Epoch 294/500
343/343          1s 2ms/step -
loss: 29454372864.0000 - val_loss: 42848735232.0000
Epoch 295/500
343/343          1s 2ms/step -
loss: 39670669312.0000 - val_loss: 42974752768.0000
Epoch 296/500
```

```
343/343             1s 2ms/step -
loss: 32645398528.0000 - val_loss: 43097051136.0000
Epoch 297/500
343/343             1s 2ms/step -
loss: 30441533440.0000 - val_loss: 42917244928.0000
Epoch 298/500
343/343             1s 2ms/step -
loss: 32181936128.0000 - val_loss: 43040559104.0000
Epoch 299/500
343/343             1s 2ms/step -
loss: 35354849280.0000 - val_loss: 42866159616.0000
Epoch 300/500
343/343             1s 3ms/step -
loss: 32324853760.0000 - val_loss: 42813546496.0000
Epoch 301/500
343/343             1s 3ms/step -
loss: 31632408576.0000 - val_loss: 42963599360.0000
Epoch 302/500
343/343             1s 4ms/step -
loss: 33323823104.0000 - val_loss: 43019563008.0000
Epoch 303/500
343/343             2s 2ms/step -
loss: 30353909760.0000 - val_loss: 42893840384.0000
Epoch 304/500
343/343             1s 2ms/step -
loss: 28603695104.0000 - val_loss: 42827206656.0000
Epoch 305/500
343/343             1s 2ms/step -
loss: 33802203136.0000 - val_loss: 42997096448.0000
Epoch 306/500
343/343             1s 2ms/step -
loss: 30066376704.0000 - val_loss: 42920460288.0000
Epoch 307/500
343/343             1s 2ms/step -
loss: 32576886784.0000 - val_loss: 42961018880.0000
Epoch 308/500
343/343             1s 2ms/step -
loss: 28226992128.0000 - val_loss: 42848362496.0000
Epoch 309/500
343/343             1s 2ms/step -
loss: 31619160064.0000 - val_loss: 42855776256.0000
Epoch 310/500
343/343             1s 2ms/step -
loss: 33680744448.0000 - val_loss: 43035000832.0000
Epoch 311/500
343/343             2s 4ms/step -
loss: 35293585408.0000 - val_loss: 42982821888.0000
Epoch 312/500
```

```
343/343             2s 2ms/step -
loss: 31999492096.0000 - val_loss: 42894135296.0000
Epoch 313/500
343/343             1s 2ms/step -
loss: 32963139584.0000 - val_loss: 42907758592.0000
Epoch 314/500
343/343             1s 2ms/step -
loss: 29877874688.0000 - val_loss: 42829336576.0000
Epoch 315/500
343/343             1s 2ms/step -
loss: 33070026752.0000 - val_loss: 43011100672.0000
Epoch 316/500
343/343             1s 2ms/step -
loss: 30097551360.0000 - val_loss: 43009454080.0000
Epoch 317/500
343/343             1s 2ms/step -
loss: 31319287808.0000 - val_loss: 42862678016.0000
Epoch 318/500
343/343             1s 2ms/step -
loss: 27873849344.0000 - val_loss: 42816638976.0000
Epoch 319/500
343/343             1s 2ms/step -
loss: 33959768064.0000 - val_loss: 42888351744.0000
Epoch 320/500
343/343             1s 2ms/step -
loss: 31233286144.0000 - val_loss: 42877337600.0000
Epoch 321/500
343/343             2s 4ms/step -
loss: 32305201152.0000 - val_loss: 43003056128.0000
Epoch 322/500
343/343             2s 2ms/step -
loss: 32776554496.0000 - val_loss: 42988400640.0000
Epoch 323/500
343/343             1s 2ms/step -
loss: 33369300992.0000 - val_loss: 43000913920.0000
Epoch 324/500
343/343             1s 2ms/step -
loss: 34936991744.0000 - val_loss: 42932727808.0000
Epoch 325/500
343/343             1s 2ms/step -
loss: 30781945856.0000 - val_loss: 42946879488.0000
Epoch 326/500
343/343             1s 2ms/step -
loss: 33092071424.0000 - val_loss: 43034230784.0000
Epoch 327/500
343/343             1s 3ms/step -
loss: 29232865280.0000 - val_loss: 42886545408.0000
Epoch 328/500
```

```
343/343              2s 4ms/step -
loss: 31975022592.0000 - val_loss: 42856226816.0000
Epoch 329/500
343/343              2s 2ms/step -
loss: 32517804032.0000 - val_loss: 42838634496.0000
Epoch 330/500
343/343              1s 3ms/step -
loss: 29417697280.0000 - val_loss: 42734596096.0000
Epoch 331/500
343/343              1s 4ms/step -
loss: 29111998464.0000 - val_loss: 42928947200.0000
Epoch 332/500
343/343              2s 2ms/step -
loss: 30051821568.0000 - val_loss: 42925334528.0000
Epoch 333/500
343/343              1s 2ms/step -
loss: 29181577216.0000 - val_loss: 42824192000.0000
Epoch 334/500
343/343              1s 2ms/step -
loss: 32301852672.0000 - val_loss: 42871881728.0000
Epoch 335/500
343/343              1s 2ms/step -
loss: 33059186688.0000 - val_loss: 42978820096.0000
Epoch 336/500
343/343              1s 2ms/step -
loss: 30308610048.0000 - val_loss: 42838687744.0000
Epoch 337/500
343/343              1s 2ms/step -
loss: 27260166144.0000 - val_loss: 42771369984.0000
Epoch 338/500
343/343              1s 2ms/step -
loss: 29234081792.0000 - val_loss: 42869170176.0000
Epoch 339/500
343/343              1s 2ms/step -
loss: 31389569024.0000 - val_loss: 42881740800.0000
Epoch 340/500
343/343              1s 2ms/step -
loss: 34534375424.0000 - val_loss: 43000893440.0000
Epoch 341/500
343/343              2s 3ms/step -
loss: 33463971840.0000 - val_loss: 42989248512.0000
Epoch 342/500
343/343              1s 4ms/step -
loss: 30957328384.0000 - val_loss: 42951380992.0000
Epoch 343/500
343/343              1s 4ms/step -
loss: 31825985536.0000 - val_loss: 42935377920.0000
Epoch 344/500
```

```
343/343              2s 2ms/step -
loss: 29278529536.0000 - val_loss: 42835517440.0000
Epoch 345/500
343/343              1s 2ms/step -
loss: 29817116672.0000 - val_loss: 42839425024.0000
Epoch 346/500
343/343              1s 2ms/step -
loss: 34808025088.0000 - val_loss: 42956541952.0000
Epoch 347/500
343/343              1s 2ms/step -
loss: 27704303616.0000 - val_loss: 42742009856.0000
Epoch 348/500
343/343              1s 2ms/step -
loss: 33053370368.0000 - val_loss: 42890072064.0000
Epoch 349/500
343/343              1s 2ms/step -
loss: 33182912512.0000 - val_loss: 42964381696.0000
Epoch 350/500
343/343              1s 2ms/step -
loss: 28856766464.0000 - val_loss: 42771578880.0000
Epoch 351/500
343/343              1s 2ms/step -
loss: 32995794944.0000 - val_loss: 42844717056.0000
Epoch 352/500
343/343              1s 2ms/step -
loss: 28633094144.0000 - val_loss: 42884366336.0000
Epoch 353/500
343/343              2s 4ms/step -
loss: 28517910528.0000 - val_loss: 42854322176.0000
Epoch 354/500
343/343              1s 4ms/step -
loss: 30894901248.0000 - val_loss: 42835419136.0000
Epoch 355/500
343/343              2s 2ms/step -
loss: 30193809408.0000 - val_loss: 42873303040.0000
Epoch 356/500
343/343              1s 2ms/step -
loss: 33332905984.0000 - val_loss: 42867904512.0000
Epoch 357/500
343/343              1s 2ms/step -
loss: 33276497920.0000 - val_loss: 42892668928.0000
Epoch 358/500
343/343              1s 2ms/step -
loss: 29555871744.0000 - val_loss: 42918236160.0000
Epoch 359/500
343/343              1s 2ms/step -
loss: 31344574464.0000 - val_loss: 42917937152.0000
Epoch 360/500
```

```
343/343             1s 2ms/step -
loss: 32036548608.0000 - val_loss: 42911014912.0000
Epoch 361/500
343/343             1s 2ms/step -
loss: 31360512000.0000 - val_loss: 42872803328.0000
Epoch 362/500
343/343             1s 2ms/step -
loss: 28450451456.0000 - val_loss: 42810290176.0000
Epoch 363/500
343/343             1s 3ms/step -
loss: 35267940352.0000 - val_loss: 42995441664.0000
Epoch 364/500
343/343             2s 4ms/step -
loss: 29012862976.0000 - val_loss: 42902892544.0000
Epoch 365/500
343/343             1s 4ms/step -
loss: 30137769984.0000 - val_loss: 42859028480.0000
Epoch 366/500
343/343             2s 2ms/step -
loss: 34259908608.0000 - val_loss: 42963378176.0000
Epoch 367/500
343/343             1s 2ms/step -
loss: 28709515264.0000 - val_loss: 42876620800.0000
Epoch 368/500
343/343             1s 2ms/step -
loss: 41316896768.0000 - val_loss: 43059015680.0000
Epoch 369/500
343/343             1s 2ms/step -
loss: 31479814144.0000 - val_loss: 42860994560.0000
Epoch 370/500
343/343             1s 2ms/step -
loss: 26056325120.0000 - val_loss: 42756608000.0000
Epoch 371/500
343/343             1s 2ms/step -
loss: 28470331392.0000 - val_loss: 42791489536.0000
Epoch 372/500
343/343             1s 2ms/step -
loss: 29107277824.0000 - val_loss: 42825523200.0000
Epoch 373/500
343/343             1s 2ms/step -
loss: 32238020608.0000 - val_loss: 42923470848.0000
Epoch 374/500
343/343             1s 4ms/step -
loss: 33617494016.0000 - val_loss: 43038081024.0000
Epoch 375/500
343/343             1s 3ms/step -
loss: 31152242688.0000 - val_loss: 43003662336.0000
Epoch 376/500
```

```
343/343             1s 4ms/step -
loss: 41465393152.0000 - val_loss: 43072622592.0000
Epoch 377/500
343/343             1s 3ms/step -
loss: 34979622912.0000 - val_loss: 43146424320.0000
Epoch 378/500
343/343             1s 2ms/step -
loss: 31187838976.0000 - val_loss: 42884100096.0000
Epoch 379/500
343/343             1s 2ms/step -
loss: 29799866368.0000 - val_loss: 42789208064.0000
Epoch 380/500
343/343             1s 2ms/step -
loss: 31371497472.0000 - val_loss: 42897252352.0000
Epoch 381/500
343/343             1s 2ms/step -
loss: 30766032896.0000 - val_loss: 42952605696.0000
Epoch 382/500
343/343             1s 2ms/step -
loss: 32833249280.0000 - val_loss: 42943217664.0000
Epoch 383/500
343/343             1s 2ms/step -
loss: 33698152448.0000 - val_loss: 42910797824.0000
Epoch 384/500
343/343             1s 2ms/step -
loss: 32317290496.0000 - val_loss: 43035144192.0000
Epoch 385/500
343/343             1s 2ms/step -
loss: 33698678784.0000 - val_loss: 43078975488.0000
Epoch 386/500
343/343             1s 3ms/step -
loss: 32112211968.0000 - val_loss: 42979119104.0000
Epoch 387/500
343/343             2s 4ms/step -
loss: 35555635200.0000 - val_loss: 43016650752.0000
Epoch 388/500
343/343             1s 4ms/step -
loss: 32935770112.0000 - val_loss: 42963701760.0000
Epoch 389/500
343/343             2s 2ms/step -
loss: 30007883776.0000 - val_loss: 42886004736.0000
Epoch 390/500
343/343             1s 2ms/step -
loss: 35895873536.0000 - val_loss: 42983833600.0000
Epoch 391/500
343/343             1s 2ms/step -
loss: 34310479872.0000 - val_loss: 43020255232.0000
Epoch 392/500
```

```
343/343            1s 2ms/step -
loss: 28892981248.0000 - val_loss: 42857787392.0000
Epoch 393/500
343/343            1s 3ms/step -
loss: 34809987072.0000 - val_loss: 43101917184.0000
Epoch 394/500
343/343            1s 2ms/step -
loss: 30101121024.0000 - val_loss: 42854060032.0000
Epoch 395/500
343/343            1s 2ms/step -
loss: 28666679296.0000 - val_loss: 42797006848.0000
Epoch 396/500
343/343            1s 3ms/step -
loss: 31177852928.0000 - val_loss: 42888372224.0000
Epoch 397/500
343/343            1s 4ms/step -
loss: 32534061056.0000 - val_loss: 42885595136.0000
Epoch 398/500
343/343            2s 2ms/step -
loss: 30988419072.0000 - val_loss: 42908082176.0000
Epoch 399/500
343/343            1s 2ms/step -
loss: 32757774336.0000 - val_loss: 42854141952.0000
Epoch 400/500
343/343            1s 2ms/step -
loss: 31598221312.0000 - val_loss: 42881921024.0000
Epoch 401/500
343/343            1s 2ms/step -
loss: 31973244928.0000 - val_loss: 43003043840.0000
Epoch 402/500
343/343            1s 2ms/step -
loss: 30010007552.0000 - val_loss: 42857279488.0000
Epoch 403/500
343/343            1s 2ms/step -
loss: 35071053824.0000 - val_loss: 42940915712.0000
Epoch 404/500
343/343            1s 2ms/step -
loss: 33226692608.0000 - val_loss: 42892214272.0000
Epoch 405/500
343/343            1s 2ms/step -
loss: 31444695040.0000 - val_loss: 42941644800.0000
Epoch 406/500
343/343            1s 2ms/step -
loss: 26769182720.0000 - val_loss: 42756120576.0000
Epoch 407/500
343/343            2s 4ms/step -
loss: 32389914624.0000 - val_loss: 42934439936.0000
Epoch 408/500
```

```
343/343             1s 4ms/step -
loss: 29795641344.0000 - val_loss: 42929799168.0000
Epoch 409/500
343/343             1s 3ms/step -
loss: 28175081472.0000 - val_loss: 42855645184.0000
Epoch 410/500
343/343             1s 2ms/step -
loss: 34493161472.0000 - val_loss: 42983235584.0000
Epoch 411/500
343/343             1s 2ms/step -
loss: 32284659712.0000 - val_loss: 42869956608.0000
Epoch 412/500
343/343             1s 2ms/step -
loss: 28266371072.0000 - val_loss: 42780229632.0000
Epoch 413/500
343/343             1s 2ms/step -
loss: 34402115584.0000 - val_loss: 42982404096.0000
Epoch 414/500
343/343             1s 2ms/step -
loss: 33546962944.0000 - val_loss: 43017519104.0000
Epoch 415/500
343/343             1s 2ms/step -
loss: 31684163584.0000 - val_loss: 42938281984.0000
Epoch 416/500
343/343             1s 2ms/step -
loss: 32990320640.0000 - val_loss: 42929573888.0000
Epoch 417/500
343/343             1s 2ms/step -
loss: 28832034816.0000 - val_loss: 42844049408.0000
Epoch 418/500
343/343             1s 2ms/step -
loss: 30691528704.0000 - val_loss: 42838519808.0000
Epoch 419/500
343/343             1s 2ms/step -
loss: 35741343744.0000 - val_loss: 42920148992.0000
Epoch 420/500
343/343             2s 4ms/step -
loss: 33841606656.0000 - val_loss: 43044257792.0000
Epoch 421/500
343/343             2s 2ms/step -
loss: 29293625344.0000 - val_loss: 42882338816.0000
Epoch 422/500
343/343             1s 3ms/step -
loss: 38866219008.0000 - val_loss: 42880778240.0000
Epoch 423/500
343/343             1s 2ms/step -
loss: 35930132480.0000 - val_loss: 42953601024.0000
Epoch 424/500
```

```
343/343              1s 2ms/step -
loss: 30947850240.0000 - val_loss: 42974093312.0000
Epoch 425/500
343/343              1s 2ms/step -
loss: 35609018368.0000 - val_loss: 42909704192.0000
Epoch 426/500
343/343              1s 2ms/step -
loss: 34856452096.0000 - val_loss: 43011334144.0000
Epoch 427/500
343/343              1s 2ms/step -
loss: 32472731648.0000 - val_loss: 43000471552.0000
Epoch 428/500
343/343              1s 2ms/step -
loss: 30606297088.0000 - val_loss: 42960355328.0000
Epoch 429/500
343/343              1s 2ms/step -
loss: 28775391232.0000 - val_loss: 42860732416.0000
Epoch 430/500
343/343              1s 4ms/step -
loss: 35214000128.0000 - val_loss: 42915409920.0000
Epoch 431/500
343/343              2s 3ms/step -
loss: 29622392832.0000 - val_loss: 42761404416.0000
Epoch 432/500
343/343              1s 2ms/step -
loss: 28359606272.0000 - val_loss: 42750464000.0000
Epoch 433/500
343/343              1s 2ms/step -
loss: 28545355776.0000 - val_loss: 42786435072.0000
Epoch 434/500
343/343              1s 2ms/step -
loss: 37507596288.0000 - val_loss: 42960793600.0000
Epoch 435/500
343/343              1s 2ms/step -
loss: 29595781120.0000 - val_loss: 42866098176.0000
Epoch 436/500
343/343              1s 2ms/step -
loss: 34117775360.0000 - val_loss: 42912272384.0000
Epoch 437/500
343/343              1s 2ms/step -
loss: 35571392512.0000 - val_loss: 43078320128.0000
Epoch 438/500
343/343              1s 2ms/step -
loss: 27161159680.0000 - val_loss: 42851221504.0000
Epoch 439/500
343/343              1s 2ms/step -
loss: 31806169088.0000 - val_loss: 42949107712.0000
Epoch 440/500
```

```
343/343              1s 2ms/step -
loss: 33644984320.0000 - val_loss: 42901196800.0000
Epoch 441/500
343/343              2s 4ms/step -
loss: 33911220224.0000 - val_loss: 42884186112.0000
Epoch 442/500
343/343              2s 3ms/step -
loss: 36443762688.0000 - val_loss: 42969870336.0000
Epoch 443/500
343/343              1s 2ms/step -
loss: 27774332928.0000 - val_loss: 42866798592.0000
Epoch 444/500
343/343              1s 2ms/step -
loss: 35143589888.0000 - val_loss: 42879873024.0000
Epoch 445/500
343/343              1s 2ms/step -
loss: 29468702720.0000 - val_loss: 42844352512.0000
Epoch 446/500
343/343              1s 2ms/step -
loss: 28949049344.0000 - val_loss: 42810343424.0000
Epoch 447/500
343/343              1s 2ms/step -
loss: 33106087936.0000 - val_loss: 42944618496.0000
Epoch 448/500
343/343              1s 2ms/step -
loss: 35574874112.0000 - val_loss: 42936786944.0000
Epoch 449/500
343/343              1s 2ms/step -
loss: 35782316032.0000 - val_loss: 42916425728.0000
Epoch 450/500
343/343              2s 4ms/step -
loss: 30671585280.0000 - val_loss: 42826805248.0000
Epoch 451/500
343/343              2s 2ms/step -
loss: 30007105536.0000 - val_loss: 42905214976.0000
Epoch 452/500
343/343              1s 2ms/step -
loss: 30275682304.0000 - val_loss: 42885480448.0000
Epoch 453/500
343/343              1s 2ms/step -
loss: 33725843456.0000 - val_loss: 42938867712.0000
Epoch 454/500
343/343              1s 2ms/step -
loss: 31230738432.0000 - val_loss: 42881523712.0000
Epoch 455/500
343/343              1s 2ms/step -
loss: 31386310656.0000 - val_loss: 42928340992.0000
Epoch 456/500
```

```
343/343              1s 2ms/step -
loss: 30885654528.0000 - val_loss: 42878631936.0000
Epoch 457/500
343/343              1s 2ms/step -
loss: 31269076992.0000 - val_loss: 42881581056.0000
Epoch 458/500
343/343              1s 2ms/step -
loss: 29719373824.0000 - val_loss: 42906906624.0000
Epoch 459/500
343/343              1s 2ms/step -
loss: 32377059328.0000 - val_loss: 43007434752.0000
Epoch 460/500
343/343              2s 4ms/step -
loss: 30918758400.0000 - val_loss: 42927460352.0000
Epoch 461/500
343/343              2s 2ms/step -
loss: 31246456832.0000 - val_loss: 42886893568.0000
Epoch 462/500
343/343              1s 2ms/step -
loss: 29056745472.0000 - val_loss: 42908327936.0000
Epoch 463/500
343/343              1s 2ms/step -
loss: 31126419456.0000 - val_loss: 42983882752.0000
Epoch 464/500
343/343              1s 2ms/step -
loss: 29713803264.0000 - val_loss: 42947461120.0000
Epoch 465/500
343/343              1s 2ms/step -
loss: 36948914176.0000 - val_loss: 43036598272.0000
Epoch 466/500
343/343              1s 3ms/step -
loss: 38749757440.0000 - val_loss: 43098484736.0000
Epoch 467/500
343/343              1s 2ms/step -
loss: 28217276416.0000 - val_loss: 42886705152.0000
Epoch 468/500
343/343              1s 2ms/step -
loss: 29662791680.0000 - val_loss: 42884345856.0000
Epoch 469/500
343/343              1s 3ms/step -
loss: 32277770240.0000 - val_loss: 42918047744.0000
Epoch 470/500
343/343              1s 4ms/step -
loss: 28777689088.0000 - val_loss: 42777792512.0000
Epoch 471/500
343/343              1s 4ms/step -
loss: 33669326848.0000 - val_loss: 43002032128.0000
Epoch 472/500
```

```
343/343              2s 2ms/step -
loss: 30191343616.0000 - val_loss: 42980655104.0000
Epoch 473/500
343/343              1s 2ms/step -
loss: 31333099520.0000 - val_loss: 42937696256.0000
Epoch 474/500
343/343              1s 2ms/step -
loss: 34945638400.0000 - val_loss: 42961076224.0000
Epoch 475/500
343/343              1s 2ms/step -
loss: 31062650880.0000 - val_loss: 42935103488.0000
Epoch 476/500
343/343              1s 2ms/step -
loss: 29183422464.0000 - val_loss: 42885365760.0000
Epoch 477/500
343/343              1s 2ms/step -
loss: 31819241472.0000 - val_loss: 42894848000.0000
Epoch 478/500
343/343              1s 2ms/step -
loss: 30054100992.0000 - val_loss: 42898087936.0000
Epoch 479/500
343/343              1s 2ms/step -
loss: 31551283200.0000 - val_loss: 42882715648.0000
Epoch 480/500
343/343              2s 4ms/step -
loss: 31068321792.0000 - val_loss: 42859208704.0000
Epoch 481/500
343/343              2s 2ms/step -
loss: 36994195456.0000 - val_loss: 42862325760.0000
Epoch 482/500
343/343              1s 2ms/step -
loss: 28307316736.0000 - val_loss: 42855759872.0000
Epoch 483/500
343/343              1s 2ms/step -
loss: 28863455232.0000 - val_loss: 42812477440.0000
Epoch 484/500
343/343              1s 2ms/step -
loss: 31409588224.0000 - val_loss: 42986135552.0000
Epoch 485/500
343/343              1s 2ms/step -
loss: 31305795584.0000 - val_loss: 42919944192.0000
Epoch 486/500
343/343              1s 2ms/step -
loss: 26453405696.0000 - val_loss: 42795302912.0000
Epoch 487/500
343/343              1s 2ms/step -
loss: 29597636608.0000 - val_loss: 42867056640.0000
Epoch 488/500
```

```
343/343              1s 2ms/step -
loss: 35627692032.0000 - val_loss: 43063504896.0000
Epoch 489/500
343/343              1s 3ms/step -
loss: 34915373056.0000 - val_loss: 43024240640.0000
Epoch 490/500
343/343              2s 4ms/step -
loss: 32213311488.0000 - val_loss: 42900754432.0000
Epoch 491/500
343/343              1s 4ms/step -
loss: 30130882560.0000 - val_loss: 42807492608.0000
Epoch 492/500
343/343              2s 2ms/step -
loss: 30525609984.0000 - val_loss: 42868342784.0000
Epoch 493/500
343/343              1s 2ms/step -
loss: 35176796160.0000 - val_loss: 42922201088.0000
Epoch 494/500
343/343              1s 2ms/step -
loss: 30684151808.0000 - val_loss: 42884325376.0000
Epoch 495/500
343/343              1s 2ms/step -
loss: 29816803328.0000 - val_loss: 42906611712.0000
Epoch 496/500
343/343              1s 2ms/step -
loss: 35033673728.0000 - val_loss: 42911264768.0000
Epoch 497/500
343/343              1s 2ms/step -
loss: 28330539008.0000 - val_loss: 42818789376.0000
Epoch 498/500
343/343              1s 2ms/step -
loss: 29398165504.0000 - val_loss: 42817953792.0000
Epoch 499/500
343/343              1s 2ms/step -
loss: 30783709184.0000 - val_loss: 42868314112.0000
Epoch 500/500
343/343              1s 2ms/step -
loss: 31868176384.0000 - val_loss: 42886836224.0000
```

[ ]:
```python
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```python
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R2): {r2}")
```

```
115/115              0s 2ms/step
Mean Absolute Error (MAE): 101901.49150833985
Mean Squared Error (MSE): 42886843677.44978
R-squared (R2): 0.7202107728982436
```