Kaggle_1

September 13, 2021

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import hvplot.pandas
%matplotlib inline
```

[3]:
```
housing_dataset = pd.read_csv("/home/ai/Downloads/Datasets/USA_Housing.csv")
```

[4]:
```
housing_dataset.info()
```

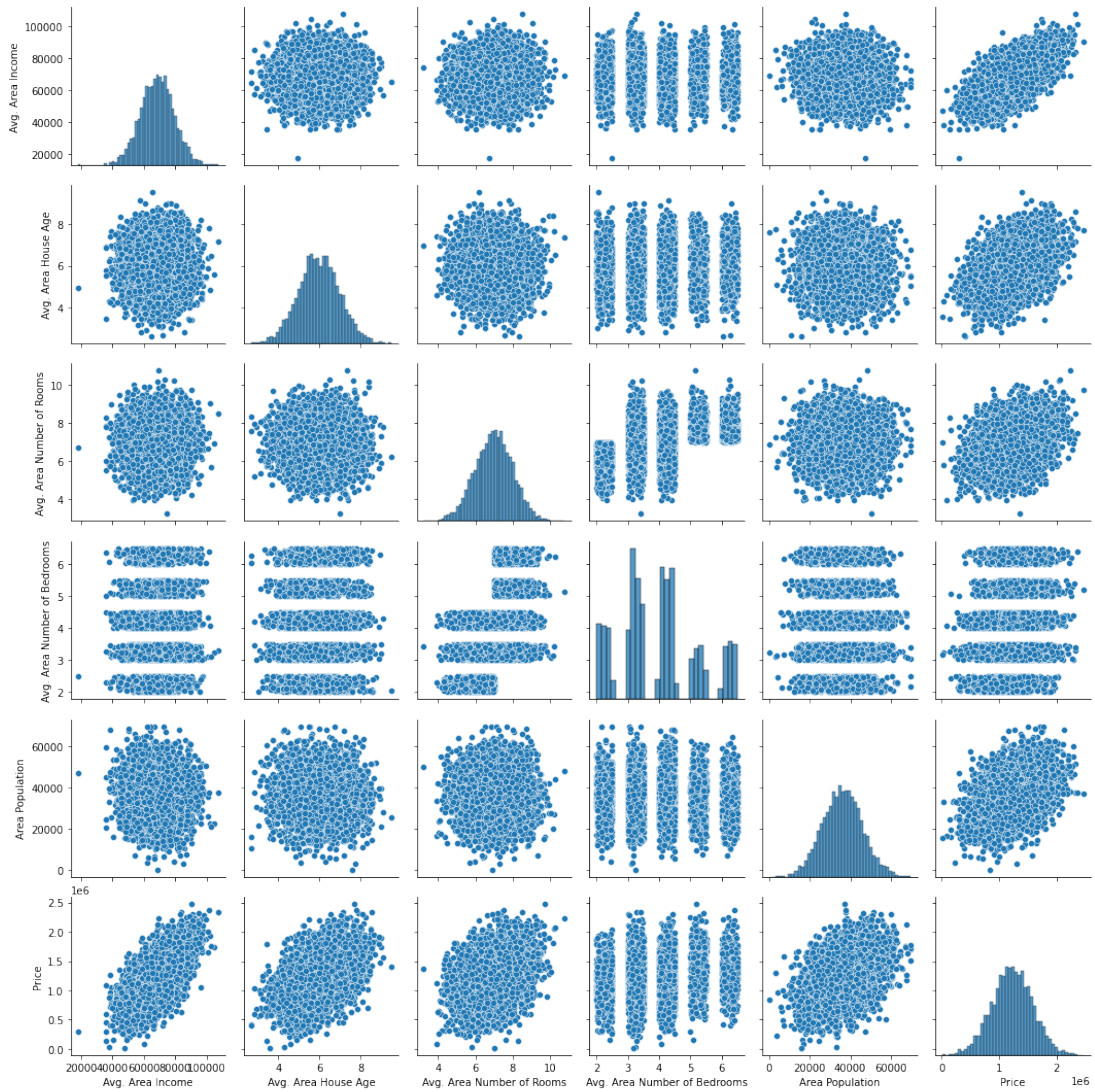
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

[5]:
```
housing_dataset.columns
```

[5]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

[6]:
```
sns.pairplot(housing_dataset)
```

[6]:
```
<seaborn.axisgrid.PairGrid at 0x7f250465c9d0>
```



[7]:
```
housing_dataset.hvplot.hist(by='Price', subplots=False, width=1000)
```

[7]:
```
:NdOverlay   [Element]
   :Histogram   [Area Population]   (Area Population_count)
```

[8]:
```
housing_dataset.hvplot.hist("Price")
```

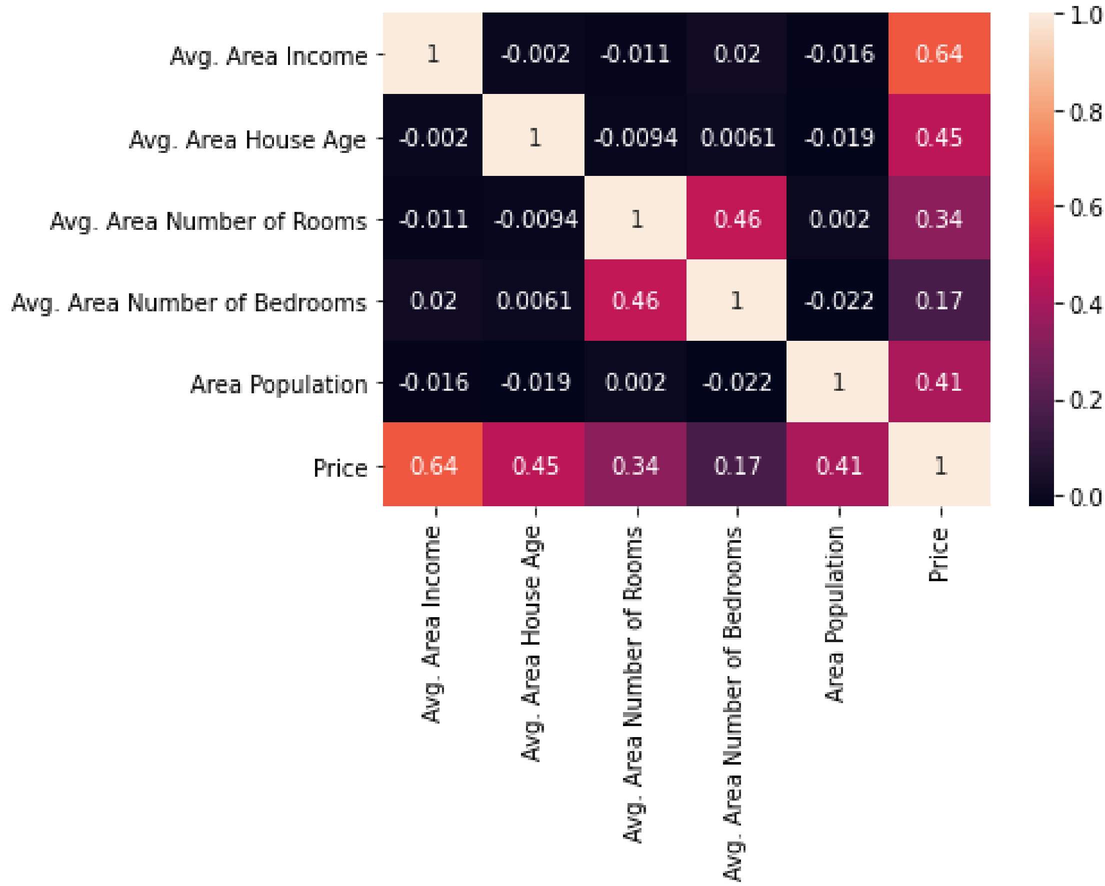
[8]:
```
:Histogram   [Price]   (Price_count)
```

[9]:
```
housing_dataset.hvplot.scatter(x='Avg. Area House Age', y = 'Price')
```

[9]:
```
:Scatter   [Avg. Area House Age]   (Price)
```

[10]:
```
sns.heatmap(housing_dataset.corr(), annot=True)
```

[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2500bdc1d0>



[11]:
```
X = housing_dataset.iloc[:, 0:5]
y = housing_dataset.iloc[:, 5:6]
len(X)
```

[11]: 5000

[12]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, )
```

[13]:
```
from sklearn import metrics
from sklearn.model_selection import cross_val_score

def cross_val(model):
    pred = cross_val_score(model, X, y, cv=10)
```

```
    return pred.mean()

def print_evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    print('MAE:', mae)
    print('MSE:', mse)
    print('RMSE:', rmse)
    print('R2 Square', r2_square)
    print('__________________________________')

def evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    return mae, mse, rmse, r2_square
```

[14]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('std_scalar', StandardScaler())
])

X_train = pipeline.fit_transform(X_train)
X_test = pipeline.transform(X_test)
```

[15]:
```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression(normalize=True)
lin_reg.fit(X_train, y_train)
```

[15]:
```
LinearRegression(normalize=True)
```

[16]:
```
print(lin_reg.intercept_)
```

```
[1233555.4251904]
```

[17]:
```
from collections import Iterable
def flatten(lis):
     for item in lis:
            if isinstance(item, Iterable) and not isinstance(item, str):
                for x in flatten(item):
                    yield x
```

```
        else:
            yield item
```

```
/home/ai/anaconda3/envs/gputest/lib/python3.7/site-
packages/ipykernel_launcher.py:1: DeprecationWarning: Using or importing the
ABCs from 'collections' instead of from 'collections.abc' is deprecated since
Python 3.3,and in 3.9 it will stop working
  """Entry point for launching an IPython kernel.
```

[20]:

```
pd.DataFrame( list(flatten(lin_reg.coef_)), X.columns, columns=['Coefficient'])
```

[20]:

	Coefficient
Avg. Area Income	231500.026152
Avg. Area House Age	163269.456192
Avg. Area Number of Rooms	121568.596453
Avg. Area Number of Bedrooms	1751.847967
Area Population	151653.388548

[21]:

```
pred = lin_reg.predict(X_test)
```

[39]:

```
type(y_test)
```

[39]:

```
pandas.core.frame.DataFrame
```

[50]:

```
y_test.values.tolist()
```

[50]:

```
[[606863.141473895],
 [1510659.22817097],
 [1173474.37883732],
 [742449.31063762],
 [1433260.22774188],
 [1170169.25920771],
 [1040781.79535934],
 [1232004.82983475],
 [1128842.20902835],
 [1472967.81431064],
 [1388209.21929933],
 [1359052.84576468],
 [1174779.15776156],
 [1306056.56543002],
 [1583897.53776365],
 [1127873.57747254],
 [1137801.93271183],
 [1179987.3238175],
 [1440317.76310708],
 [1387701.11105296],
```

```
[1124719.49016535],
[981753.238408371],
[1198656.87240769],
[1366854.24050653],
[1279353.92451211],
[1365151.39422781],
[1306288.2274582],
[1822438.41990484],
[885796.279596747],
[730739.418950083],
[1907859.05893151],
[1588196.23290249],
[795507.8434415429],
[1008712.57507319],
[1003266.87623527],
[674657.910663075],
[626085.933004653],
[1463130.27297894],
[1409238.79759642],
[977024.1101963429],
[2024121.46147324],
[1071537.37638207],
[675536.391488567],
[1348221.76524399],
[1384466.43691295],
[1100152.08776636],
[1345962.95498853],
[962581.688327075],
[1538039.5989438999],
[758262.612110653],
[449728.10099925],
[1270869.78317193],
[1618721.13842984],
[253185.701508582],
[1281741.16338047],
[1616382.93211806],
[1057838.49342254],
[1747004.58275903],
[1275497.16062662],
[1668905.28457987],
[1411217.49285109],
[1088222.42090221],
[879062.5907073609],
[1492011.49586213],
[1554301.93808174],
[999369.798364609],
[1555497.54638866],
```

```
[1445804.83165148],
[1602903.28070873],
[837540.1697407691],
[962531.664889236],
[1227568.95802279],
[1331656.4939736],
[586381.547325308],
[1127248.61023368],
[1379671.07478299],
[541953.905680242],
[1234531.60050334],
[945981.5296322841],
[1300031.23033322],
[717682.085703176],
[1741959.83443074],
[1412449.47858682],
[953939.3299056441],
[1612102.09670813],
[1104701.23442073],
[1370274.38731202],
[1197307.14600813],
[1137685.11628548],
[1191968.52004988],
[1500962.61511429],
[916140.221249743],
[1571464.96522196],
[1362239.05354028],
[1030016.64474952],
[1043628.37035454],
[1877941.97142307],
[1162949.67898603],
[968360.5234031],
[1474417.27172918],
[1594089.24144713],
[932979.360621325],
[1489648.01766795],
[843650.827890095],
[1300389.35082021],
[1558196.96857136],
[1697501.60460523],
[995849.775109822],
[1338044.38331259],
[1687998.93296047],
[1389047.55777647],
[1517152.58462546],
[1204753.00205501],
[1165877.66861295],
```

```
[1234629.59549004],
[1172413.19696021],
[1630438.85423015],
[1110931.82570237],
[1250882.29175707],
[1499988.87952261],
[1131154.51836542],
[1361872.29101591],
[1222799.33972217],
[1364194.93981793],
[1298980.82895],
[1029273.27859557],
[1448573.84467799],
[1705276.18956748],
[311111.200586772],
[895600.423741341],
[1062877.8829939002],
[568842.535575364],
[1195897.74052292],
[981559.81167832],
[1203089.9890177],
[1644242.02414954],
[493350.031167589],
[1115013.14745176],
[1450393.53041457],
[1561313.83763862],
[1757395.78723971],
[828127.0276465779],
[1395218.58026292],
[1360329.35860971],
[1027427.8619974001],
[1199760.29171634],
[1239309.91470996],
[1123753.73420143],
[1291759.39266068],
[1084255.67827228],
[1969256.93301589],
[1288069.22644892],
[814784.2457280059],
[1510883.77735139],
[2330289.70068344],
[757943.222820573],
[1825071.45225254],
[1435423.70775328],
[1285098.69079136],
[964596.749839928],
[1457767.53264774],
```

```
[999146.8693743941],
[1361007.56184712],
[1420070.02039984],
[963531.915230982],
[1116487.35709413],
[1423125.54126977],
[1010477.39216786],
[533386.771238809],
[764860.462688998],
[1000842.17761448],
[1289082.363025],
[1002690.89930323],
[1530294.57733993],
[1255202.10983879],
[353240.05467269203],
[1186441.75644398],
[1442632.54122656],
[1253370.14892134],
[1048639.78949745],
[1772780.14260328],
[935488.606494803],
[1479295.57093729],
[792174.1738214791],
[1272719.63189395],
[1603046.17935713],
[1372806.61636634],
[1731437.41599478],
[947355.9700864509],
[1107159.78060591],
[1642478.73565644],
[1530729.57845363],
[1197073.44481501],
[1347622.4830191003],
[1286974.69672285],
[1499243.16756539],
[1553459.36448355],
[891661.6959126609],
[1152145.9120119],
[1599851.89450841],
[1482617.72862202],
[1334952.60511867],
[673225.111889516],
[1159953.5929205],
[1257376.17368852],
[904497.317088796],
[1204103.086599],
[1138036.59504191],
```

```
[1228323.22490594],
[1212939.95351759],
[776125.597113738],
[1223777.49762374],
[1477595.99924808],
[1547127.38645843],
[966084.4204303529],
[984349.614154876],
[1237902.82007375],
[1172133.48684149],
[2050594.04550631],
[912422.146420949],
[1057308.2897955999],
[1308983.63216408],
[1047689.26574636],
[756216.5757576079],
[1967384.77782106],
[1430796.76256979],
[1567859.07278121],
[744741.9092512471],
[954746.576432824],
[1574919.9063463002],
[1296432.78464476],
[1085101.32400571],
[1085218.85870708],
[1080624.49716572],
[1735826.00520535],
[1626368.66731315],
[846095.967314711],
[1283764.78775609],
[1006885.8444126999],
[1329800.04619049],
[677277.596588513],
[1358213.90612806],
[890887.6807315941],
[1214986.88571544],
[1409439.07587247],
[1296610.53228993],
[528485.2467305961],
[1677204.24621443],
[1007856.26624004],
[1357710.96492256],
[1299126.82985549],
[1584213.95748502],
[1421518.60190753],
[1657667.33216395],
[1195986.29883487],
```

```
[776906.326934832],
[1414259.77535829],
[1043281.2900747],
[1031981.28292294],
[997827.253775207],
[1789607.52613989],
[556839.637132389],
[729989.349928175],
[1785449.72016789],
[868314.4735625429],
[1655466.73582652],
[926880.850551317],
[1335830.79241959],
[1456969.57682195],
[1560805.07079272],
[1053966.42543393],
[1151211.90867289],
[810741.518507977],
[1388596.243304],
[1425114.71776615],
[390948.449994154],
[1383508.06161512],
[1745516.17062634],
[1015233.0322629999],
[829128.65683814],
[717273.185257682],
[1421135.8149541],
[1053338.5923718],
[1096173.16778533],
[1603944.37993747],
[1813043.92613556],
[1186357.17904467],
[1094792.0032017],
[972536.6627723591],
[1103072.43866161],
[1210573.85526575],
[910211.018936968],
[576387.17522213],
[1309829.9356108],
[1063743.42416582],
[1006687.38713887],
[697656.592436029],
[1530124.0163168998],
[716316.4881444591],
[991934.0653650309],
[1040607.31205457],
[1496539.31289263],
```

```
[1669127.07812649],
[1557794.02127226],
[1439892.63249514],
[895845.56548697],
[1385978.98411983],
[1311389.57582794],
[816813.5323512609],
[1412242.75472738],
[1535946.64311382],
[615568.113378455],
[1447671.67468216],
[953819.209553696],
[842985.8528549679],
[1816217.00697668],
[1273518.73676148],
[1393100.42260616],
[1113314.69887382],
[1429951.80389933],
[1077805.57772632],
[1517597.52709047],
[1299430.1759553999],
[1460450.98147615],
[1375057.08346865],
[1322439.10755647],
[658644.617000323],
[1166750.31285964],
[577104.814915664],
[1261491.42538357],
[836235.0305323859],
[923246.862074019],
[883021.896376765],
[1545864.1664834998],
[1495390.49870273],
[1003905.05954605],
[780934.037963336],
[1418977.82858253],
[1340343.85651884],
[642809.966743474],
[1890056.38519368],
[889385.90158426],
[1940933.47241606],
[1378269.57282251],
[999970.67942849],
[639471.73613359],
[840682.60887661],
[931768.04553571],
[1930805.94669456],
```

```
[1159596.51861962],
[1108406.78591683],
[1775531.40394545],
[1369970.61328254],
[401148.568791379],
[786257.782646887],
[1129582.83842209],
[1200401.98894193],
[1860358.46831207],
[832536.885367347],
[944001.5583308409],
[1441882.52170551],
[1444572.97149721],
[991892.317484247],
[1279803.9500805],
[1062206.32094414],
[1260826.63799718],
[1820439.35291089],
[1729061.65557134],
[1800685.92248892],
[968045.477835415],
[1729559.80913131],
[1361219.30037656],
[302355.835978956],
[963934.013010864],
[1409038.98711763],
[1169547.52118279],
[1341069.15980991],
[1572099.68985623],
[2001910.21599521],
[1629345.78653286],
[1838092.10249886],
[1438290.85142595],
[1472887.24706053],
[1676172.30485552],
[470008.138227564],
[596350.350685955],
[1197975.08547024],
[1546408.55841364],
[1119992.61883545],
[710269.2139681029],
[1705889.88669841],
[1845325.3567773998],
[1056984.18175283],
[961495.096445765],
[1069827.80663915],
[1073355.78370383],
```

```
[1137446.99339696],
[1032663.00307792],
[1072704.89021544],
[1414495.34313381],
[1240754.93204638],
[1166336.25929784],
[965074.804743782],
[911071.8335530659],
[1762214.67958189],
[783818.6820390391],
[948788.2757094479],
[1739761.3726113997],
[1219637.36651886],
[1819900.63664371],
[1524845.42190295],
[1813555.32993929],
[764955.498207082],
[1093344.57650394],
[1178846.06581529],
[785399.1395370429],
[945252.191803817],
[969678.567288796],
[1236874.809188],
[1414286.72449432],
[1079158.65285226],
[1196064.32994419],
[1191261.86867965],
[1928728.79155594],
[1424389.48332932],
[1409943.30299484],
[1064939.52251981],
[1672445.34797165],
[1383031.33236644],
[1390200.22429845],
[671661.713440961],
[1154917.39361164],
[31140.517620186],
[1596180.56843652],
[1140579.61789309],
[1509962.02177698],
[826587.39194003],
[491907.794371219],
[1033450.13928111],
[1086829.24563268],
[1328658.91107238],
[1376969.92880029],
[1617114.94397038],
```

```
[1708631.36537014],
[940162.724477794],
[1165253.03271806],
[1244440.13470232],
[781482.583221976],
[575694.1732158379],
[494742.543577691],
[893215.236639428],
[1113936.63783162],
[738030.649864538],
[1121023.43701055],
[1544379.74773568],
[1482123.62468682],
[885611.578262467],
[1252733.09655161],
[1490718.33494934],
[774009.547594125],
[1501303.96414075],
[884015.2270688241],
[1562573.06108102],
[1555490.62086308],
[1797621.2204361],
[1088834.29008834],
[924346.164802417],
[1569600.44563595],
[1222344.20849926],
[858088.718786957],
[954114.543953548],
[1077059.1559604001],
[1358526.74126213],
[780608.993517457],
[1537897.11811872],
[994897.7277644679],
[1440897.41242886],
[1114957.5968695],
[1238406.18978396],
[753652.27184518],
[942838.1647218129],
[879511.1964474791],
[949892.531370182],
[1156329.27404976],
[2271112.74382772],
[923508.125650491],
[1363106.12528245],
[1078016.9396491],
[1635092.43276192],
[1146480.21353497],
```

```
[1079951.15543165],
[1431517.43889219],
[1966139.92775068],
[684257.111066533],
[1016871.41491422],
[1455556.46564255],
[1222726.56436107],
[1411749.10721655],
[1409439.06376517],
[997239.707755351],
[1301486.6855461001],
[1308465.24004301],
[630943.48933854],
[1647278.68498366],
[1095727.67258837],
[1413729.75740522],
[1117741.75732422],
[1795428.7743898],
[1053891.64486387],
[1454834.98247726],
[1334537.42355949],
[860804.4460473291],
[1167119.10883666],
[1704126.8763505],
[1557276.06443164],
[1244630.9976],
[1434575.10990828],
[1566567.8831387],
[1038330.03895894],
[1205114.23020705],
[1576286.21421499],
[1574279.81397754],
[1153029.61571986],
[944288.911013619],
[1613814.87549154],
[1453327.91633352],
[1471031.42057582],
[1309938.76715168],
[324981.992992685],
[1643291.72247874],
[1919693.23049382],
[1502447.91672879],
[1435261.98681704],
[1499552.26873769],
[841276.5860989691],
[1521085.47247059],
[1340983.25035903],
```

```
[1060966.74140573],
[1190213.51290501],
[1393897.39021981],
[839426.122179982],
[1040782.48588109],
[1656785.44770054],
[553077.212612076],
[1553976.23390492],
[1496776.83013235],
[1377113.02613595],
[1295967.38741543],
[1180903.43393502],
[1789973.20520199],
[1112294.77646566],
[1748474.69409409],
[1280199.2946553999],
[1142113.38414845],
[1300479.13327003],
[1270390.24846834],
[1550181.07452172],
[1235593.27657753],
[1043415.99202097],
[1563193.09711859],
[1480227.54724344],
[1053484.8692001998],
[836883.5652618491],
[1216307.1456612],
[986285.38185614],
[1066768.00005863],
[1713351.3941515],
[1509929.83247383],
[1251430.92031862],
[1325706.90085614],
[585531.238099097],
[1194515.33809666],
[1271752.43195011],
[1642011.34461186],
[1685368.16422505],
[1214941.76172658],
[1412626.51874178],
[1650770.92968133],
[943575.130695582],
[1059870.96483225],
[1168859.1171821998],
[1346049.6558665],
[854815.1175016209],
[1028493.60145884],
```

```
[1266616.54671419],
[699222.836143786],
[1389366.7180879],
[1447492.96439529],
[1288490.26844093],
[807916.313401003],
[882666.58150817],
[1634781.27382689],
[958909.193464849],
[1603075.23661695],
[1151406.78433285],
[1661119.50062796],
[1133084.32382384],
[1523801.748313],
[1384140.22706635],
[825095.108578208],
[1917583.99861814],
[1198572.86630025],
[1693693.55312723],
[1491811.66112295],
[884891.4955195291],
[1490539.05810979],
[679228.992746839],
[1006227.53499553],
[1759180.04363531],
[1363319.8464196],
[1434144.26087246],
[1643338.99271745],
[917351.135593772],
[1151741.15459096],
[529282.084392095],
[1742566.00604802],
[1444701.32791986],
[1397461.55252241],
[1296223.70649253],
[1202992.88412315],
[1380070.4501178998],
[1597776.77050618],
[1228532.30084287],
[1456273.72393514],
[1239318.13087631],
[1514555.08213084],
[738760.747431326],
[1322010.0566818998],
[1090788.98123407],
[1899136.5354006],
[1330929.47215359],
```

```
[646198.224523062],
[1662585.93817899],
[1390990.84909224],
[1115466.58507141],
[783350.673223018],
[1376637.50562417],
[1373022.24441616],
[1640185.79829682],
[2016910.73970733],
[1408347.44473948],
[1707494.92179398],
[1481946.1940765],
[1746086.98892861],
[1074263.31723636],
[2146925.33988866],
[817460.421624823],
[566848.7320986079],
[1109059.0536268],
[1431915.97700817],
[1415073.61418119],
[873515.9385396],
[1123209.85469296],
[1336377.82525371],
[1446155.17970209],
[395901.250067375],
[1290324.64861995],
[1124396.24901271],
[1653381.312924],
[1384787.45743007],
[843633.198243946],
[950121.562106524],
[974599.958764174],
[1166158.18468767],
[1022342.57363608],
[789043.8511966041],
[1626893.54722418],
[629657.6132544071],
[1223100.54249556],
[1481146.95049889],
[1051644.55402255],
[1547133.39640045],
[747933.486938614],
[836339.816945874],
[1184614.08702402],
[832559.79755093],
[1591383.7352200001],
[859208.721224715],
```

```
[727366.947448545],
[1322782.76026948],
[1616631.33374997],
[1465120.47000627],
[735952.168889463],
[1095768.73335286],
[1719762.30804922],
[1020842.53135241],
[915992.5577704291],
[718887.2315009279],
[1498640.55065037],
[1152269.43076739],
[990894.053688713],
[1245785.05095909],
[1109639.99566839],
[1765809.4320663002],
[856931.528435322],
[613788.32198953],
[1323959.79910681],
[560598.538430964],
[813726.113831596],
[1451658.87724072],
[1566740.12792143],
[1383842.1850596],
[1011477.61428122],
[1827476.19606152],
[1246246.82775976],
[1186688.50593835],
[649223.5060818361],
[1511743.12919103],
[1291506.38576454],
[955445.853671163],
[1591269.9928074],
[1213382.22263647],
[1233965.83737812],
[1777516.4494720001],
[798073.946177208],
[1622827.29784367],
[1538903.99254514],
[1200888.77269115],
[1108204.58686276],
[454234.33443278796],
[1473071.95430455],
[1132779.83676253],
[1122563.81949877],
[1589765.4967532],
[1914509.5209888],
```

```
[839589.844483433],
[823941.922058669],
[1325149.91983277],
[949072.249595465],
[853084.152206883],
[1740719.10656986],
[909885.801415027],
[1441312.16826908],
[634218.44082844],
[727866.5251679079],
[920951.5682989659],
[1315828.85983511],
[369433.35496479203],
[657169.304161534],
[1348883.32147854],
[1440738.46889124],
[2138713.94334063],
[1664301.51232326],
[820250.3894658079],
[1650342.13064281],
[1447325.63969774],
[1000892.04640438],
[603714.788582196],
[1568181.61446469],
[1111655.90583864],
[1570904.97723406],
[869026.58199881],
[1674680.74073848],
[1129798.37272535],
[1375105.34592206],
[2045693.81974913],
[1116193.29296624],
[1246270.00968581],
[1529281.84011934],
[1220001.33423995],
[1618994.27644467],
[1689120.42652009],
[1543966.42644823],
[1370699.72237291],
[1364363.3753658999],
[1592175.51246226],
[1133814.07796292],
[482689.703375666],
[1685967.64821564],
[1399909.04869688],
[1357797.90952389],
[1031367.58851649],
```

```
[1891949.34783336],
[1065150.78675383],
[1594748.12458456],
[15938.657923287901],
[2020158.3629418],
[1010813.70905556],
[606191.866922813],
[1738788.38189407],
[846841.456423823],
[702321.752321421],
[619407.486265184],
[1476392.55075533],
[1663653.70975754],
[1443027.26306137],
[688479.292591021],
[1678347.52753798],
[1802294.28889866],
[985006.9861893159],
[1346969.10275737],
[852099.464026854],
[1507035.87371331],
[1367619.92344288],
[725040.8959547429],
[996924.6914267291],
[928210.761532312],
[1197929.7431865],
[973299.861788715],
[1167621.0464497],
[1410362.61309221],
[1643086.27789701],
[1525533.40662551],
[1191564.19993432],
[1252723.42964037],
[1468030.4077797],
[943485.047267782],
[965136.3385167391],
[1053236.59985491],
[1603627.93774969],
[864067.0615371759],
[1693384.80306549],
[1304511.89906857],
[1425632.5419742],
[1641749.65120549],
[1540418.22216255],
[987355.96092124],
[1169640.32430332],
[1127637.5573666],
```

```
[1095391.79860126],
[696467.169937912],
[1339303.6569164],
[1637649.90477094],
[577168.273943905],
[1592210.17592313],
[962501.901521562],
[1133139.92617632],
[1005431.45123382],
[717213.2687989541],
[1785374.29206076],
[2031338.3469761002],
[1580951.57856852],
[1286572.97851797],
[1308017.31423918],
[1500472.82722154],
[623181.134793923],
[1168845.81517368],
[1023162.95290663],
[1714266.27375534],
[1300133.96670156],
[1455737.13490413],
[1437053.55720162],
[2142896.55640726],
[1250866.56678963],
[1487997.87866332],
[1201110.29436287],
[1160397.82173002],
[1754969.16194479],
[1148749.93331186],
[1360920.53038608],
[1543357.45351051],
[737202.457611395],
[989643.0313890129],
[1372878.53439379],
[873509.189526672],
[628870.345640628],
[1449409.43560429],
[1355545.58790074],
[609402.775254889],
[1460341.01117644],
[1228138.1291209],
[723089.952526648],
[1418723.65698423],
[1082370.17118962],
[962069.0020449391],
[1384328.12218859],
```

```
[757469.02224512],
[1545507.56497685],
[1003642.62741729],
[1573288.81760129],
[1136448.40876046],
[1423908.25792811],
[1919053.52694713],
[1720670.87684279],
[1652845.98062876],
[1033155.32945701],
[1441376.39392555],
[1200539.36116778],
[1411756.61863455],
[1415349.64966796],
[1078534.5530804],
[1004340.03594266],
[755292.114360278],
[1280276.42437603],
[578593.5339451779],
[1085534.17857374],
[1659170.51277537],
[961727.46838822],
[1101239.49218551],
[1020098.04138186],
[1002192.58207068],
[1321888.26829997],
[1231157.25514744],
[2026303.09804233],
[973688.7554994221],
[1277780.51220943],
[1556987.72445977],
[1454699.23141047],
[1129408.87012587],
[1575966.55253915],
[1414286.72242412],
[840208.74304146],
[1012269.46867818],
[1117432.98005745],
[1550931.06784449],
[1637335.30278857],
[1795093.03400894],
[1593836.78519275],
[1051230.0005784],
[1115721.68697909],
[718650.643819188],
[896944.244271168],
[1202844.37371258],
```

```
[1811265.52321561],
[777733.6134218429],
[1906024.63648502],
[1198313.59809977],
[1473680.6037392],
[714142.24065392],
[1449033.43225425],
[1942640.26532281],
[927978.909244959],
[1455691.79650696],
[1626941.78786638],
[1235681.79295981],
[1126821.10881128],
[1311432.38811169],
[942093.924477396],
[1311902.5342297],
[1009809.98359655],
[1134142.9159604],
[762402.1788107221],
[1155513.1752],
[1230628.98596252],
[1185616.34001422],
[527749.403864781],
[1567745.80674512],
[1261843.84432174],
[1105737.46154385],
[879356.366048286],
[1337973.73201597],
[1214320.57436406],
[1469354.78233224],
[1354077.49823771],
[1411730.47725883],
[1076334.10254553],
[1629049.62351924],
[1302347.57350905],
[1366668.18528835],
[907462.451202018],
[1345004.11975651],
[1243918.8107245],
[1273868.06365871],
[1238929.11006833],
[759044.68799078],
[1760880.44750733],
[1527676.37397527],
[1287325.38185667],
[1131040.70232425],
[1081578.61947582],
```

```
 [1040376.99516961],
 [625880.7320537639],
 [1288426.69390285],
 [1637259.99866779],
 [985283.881557835],
 [852590.3528209791],
 [1086447.51066324],
 [1075723.50991438],
 [1851065.49687777],
 [1124125.92064897],
 [1860681.94702102],
 [890481.7298329041],
 [1212276.35749966],
 [1524106.96472147],
 [1838503.99427185],
 [1152798.57896721],
 [1147685.30055025],
 [1221827.82199886],
 [1536143.59193863],
 [1335904.50091554],
 [728402.000075194],
 [544320.688146663],
 [1375215.06159392],
 [1088699.39928965],
 [1409033.07088753],
 [1381430.63018879],
 [1381442.75845971],
 [1721005.41950458],
 [854487.5970384091],
 [1023965.40341073],
 [615144.898641351],
 [845890.240512914],
 [624432.965994422],
 [1543801.6545593],
 [1117011.10618471],
 [1204117.41865411],
 [1109665.5634039],
 [714148.414752902],
 [1791307.24109703],
 [891468.213279107],
 …]
```

[53]:
```
pd.DataFrame({'True Values': list(flatten(y_test))})
```

[53]:

	True Values
0	Price

[54]:
```
pd.DataFrame({'True Values': y_test.values.tolist(), 'Predicted Values':␣
↪list(flatten(pred))}).hvplot.scatter(
    x='True Values', y='Predicted Values')
```

[54]:
```
:Scatter   [True Values]   (Predicted Values)
```

[58]:
```
sol = np.array(y_test.values.tolist()) - np.array(list(flatten(pred)))
```

[60]:
```
sol
```

[60]:
```
array([[ 1.08256340e+04, -8.63986249e+05, -4.60225345e+05, …,
        -4.80311513e+05, -3.71033963e+05, -9.04158878e+05],
       [ 9.14621721e+05,  3.98098381e+04,  4.43570742e+05, …,
         4.23484573e+05,  5.32762124e+05, -3.62791416e+02],
       [ 5.77436871e+05, -2.97375011e+05,  1.06385893e+05, …,
         8.62997240e+04,  1.95577274e+05, -3.37547641e+05],
       …,
       [ 4.65514514e+05, -4.09297368e+05, -5.53646415e+03, …,
        -2.56226329e+04,  8.36549175e+04, -4.49469998e+05],
       [ 3.49795682e+05, -5.25016201e+05, -1.21255297e+05, …,
        -1.41341466e+05, -3.20639152e+04, -5.65188830e+05],
       [ 8.16463738e+05, -5.83481445e+04,  3.45412759e+05, …,
         3.25326591e+05,  4.34604141e+05, -9.85207741e+04]])
```

[62]:
```
pd.DataFrame({'Error Values': list(flatten(sol))}).hvplot.kde()
```

[62]:
```
:Distribution   [Error Values]   (Density)
```

[63]:
```
test_pred = lin_reg.predict(X_test)
train_pred = lin_reg.predict(X_train)

print('Test set evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
print('Train set evaluation:\n_____________________________________')
print_evaluate(y_train, train_pred)
```

```
Test set evaluation:
_____________________________________
MAE: 81528.47728239739
MSE: 10022058835.578112
RMSE: 100110.23342085519
R2 Square 0.9219960572792769
__________________________________
Train set evaluation:
_____________________________________
MAE: 81322.83077092048
MSE: 10294345335.798096
```

```
RMSE: 101461.05329533148
R2 Square 0.9165685616082142
__________________________________
```

[64]:
```
results_df = pd.DataFrame(data=[["Linear Regression", *evaluate(y_test,␣
↪test_pred) , cross_val(LinearRegression())]],
                          columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2 Square',␣
↪"Cross Validation"])
results_df
```

[64]:
```
               Model           MAE           MSE           RMSE  R2 Square  \
0  Linear Regression  81528.477282  1.002206e+10  100110.233421   0.921996

   Cross Validation
0          0.917379
```

[65]:
```
from sklearn.linear_model import RANSACRegressor

model = RANSACRegressor(base_estimator=LinearRegression(), max_trials=100)
model.fit(X_train, y_train)

test_pred = model.predict(X_test)
train_pred = model.predict(X_train)

print('Test set evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
print('====================================')
print('Train set evaluation:\n_____________________________________')
print_evaluate(y_train, train_pred)
```

```
Test set evaluation:
_____________________________________
MAE: 84290.81855671547
MSE: 10750719912.971506
RMSE: 103685.67843714727
R2 Square 0.9163247238859785
__________________________________
====================================
Train set evaluation:
_____________________________________
MAE: 82822.92967306884
MSE: 10759679098.102133
RMSE: 103728.87302049575
R2 Square 0.9127972226978824
__________________________________
```

[66]:
```
results_df_2 = pd.DataFrame(data=[["Robust Regression", *evaluate(y_test,␣
↪test_pred) , cross_val(RANSACRegressor())]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2␣
↪Square', "Cross Validation"])
results_df = results_df.append(results_df_2, ignore_index=True)
results_df
```

[66]:
```
               Model           MAE           MSE           RMSE  R2 Square  \
0  Linear Regression  81528.477282  1.002206e+10  100110.233421   0.921996
1  Robust Regression  84290.818557  1.075072e+10  103685.678437   0.916325

   Cross Validation
0          0.917379
1          0.909424
```

[67]:
```
from sklearn.linear_model import Ridge

model = Ridge(alpha=100, solver='cholesky', tol=0.0001, random_state=42)
model.fit(X_train, y_train)
pred = model.predict(X_test)

test_pred = model.predict(X_test)
train_pred = model.predict(X_train)

print('Test set evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
print('====================================')
print('Train set evaluation:\n_____________________________________')
print_evaluate(y_train, train_pred)
```

```
Test set evaluation:
_____________________________________
MAE: 82021.38350548885
MSE: 10092519264.127617
RMSE: 100461.53126509479
R2 Square 0.9214476478832812
__________________________________
====================================
Train set evaluation:
_____________________________________
MAE: 81686.58642106052
MSE: 10379940489.376337
RMSE: 101881.9929593858
R2 Square 0.9158748480645695
__________________________________
```

[68]:
```
results_df_2 = pd.DataFrame(data=[["Ridge Regression", *evaluate(y_test,␣
↪test_pred) , cross_val(Ridge())]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2␣
↪Square', "Cross Validation"])
results_df = results_df.append(results_df_2, ignore_index=True)
results_df
```

[68]:
```
               Model           MAE           MSE           RMSE  R2 Square  \
0  Linear Regression  81528.477282  1.002206e+10  100110.233421   0.921996
1  Robust Regression  84290.818557  1.075072e+10  103685.678437   0.916325
2   Ridge Regression  82021.383505  1.009252e+10  100461.531265   0.921448

   Cross Validation
0          0.917379
1          0.909424
2          0.917379
```

[69]:
```
from sklearn.preprocessing import PolynomialFeatures

poly_reg = PolynomialFeatures(degree=2)

X_train_2_d = poly_reg.fit_transform(X_train)
X_test_2_d = poly_reg.transform(X_test)

lin_reg = LinearRegression(normalize=True)
lin_reg.fit(X_train_2_d,y_train)

test_pred = lin_reg.predict(X_test_2_d)
train_pred = lin_reg.predict(X_train_2_d)

print('Test set evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
print('====================================')
print('Train set evaluation:\n_____________________________________')
print_evaluate(y_train, train_pred)
```

```
Test set evaluation:
_____________________________________
MAE: 81487.593488941
MSE: 10034501727.806126
RMSE: 100172.36009901197
R2 Square 0.9218992114446477
__________________________________
====================================
Train set evaluation:
_____________________________________
MAE: 81148.95768250916
```

```
MSE: 10268693780.678932
RMSE: 101334.56360333788
R2 Square 0.9167764569207151
__________________________________
```

[70]:

```
results_df_2 = pd.DataFrame(data=[["Polynomail Regression", *evaluate(y_test,␣
↪test_pred), 0]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2␣
↪Square', 'Cross Validation'])
results_df = results_df.append(results_df_2, ignore_index=True)
results_df
```

[70]:

```
                   Model           MAE           MSE           RMSE  \
0      Linear Regression  81528.477282  1.002206e+10  100110.233421
1      Robust Regression  84290.818557  1.075072e+10  103685.678437
2       Ridge Regression  82021.383505  1.009252e+10  100461.531265
3  Polynomail Regression  81487.593489  1.003450e+10  100172.360099

   R2 Square  Cross Validation
0   0.921996          0.917379
1   0.916325          0.909424
2   0.921448          0.917379
3   0.921899          0.000000
```

[71]:

```
from sklearn.linear_model import SGDRegressor

sgd_reg = SGDRegressor(n_iter_no_change=250, penalty=None, eta0=0.0001,␣
↪max_iter=100000)
sgd_reg.fit(X_train, y_train)

test_pred = sgd_reg.predict(X_test)
train_pred = sgd_reg.predict(X_train)

print('Test set evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
print('====================================')
print('Train set evaluation:\n_____________________________________')
print_evaluate(y_train, train_pred)
```

```
/home/ai/anaconda3/envs/gputest/lib/python3.7/site-
packages/sklearn/utils/validation.py:73: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  return f(**kwargs)

Test set evaluation:
_____________________________________
MAE: 81528.48983509491
```

```
MSE: 10022061953.013712
RMSE: 100110.24899086863
R2 Square 0.921996033015573
__________________________________
====================================
Train set evaluation:
_____________________________________
MAE: 81322.82658669332
MSE: 10294345335.953741
RMSE: 101461.0532960985
R2 Square 0.9165685616069528
__________________________________
```

[72]:
```
results_df_2 = pd.DataFrame(data=[["Stochastic Gradient Descent",␣
↪*evaluate(y_test, test_pred), 0]],
                            columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2␣
↪Square', 'Cross Validation'])
results_df = results_df.append(results_df_2, ignore_index=True)
results_df
```

[72]:
```
                         Model           MAE           MSE           RMSE  \
0            Linear Regression  81528.477282  1.002206e+10  100110.233421
1            Robust Regression  84290.818557  1.075072e+10  103685.678437
2             Ridge Regression  82021.383505  1.009252e+10  100461.531265
3         Polynomail Regression  81487.593489  1.003450e+10  100172.360099
4  Stochastic Gradient Descent  81528.489835  1.002206e+10  100110.248991

   R2 Square  Cross Validation
0   0.921996          0.917379
1   0.916325          0.909424
2   0.921448          0.917379
3   0.921899          0.000000
4   0.921996          0.000000
```

[73]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam

X_train = np.array(X_train)
X_test = np.array(X_test)
y_train = np.array(y_train)
y_test = np.array(y_test)

model = Sequential()

model.add(Dense(X_train.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
```

```
# model.add(Dropout(0.2))

model.add(Dense(64, activation='relu'))
# model.add(Dropout(0.2))

model.add(Dense(128, activation='relu'))
# model.add(Dropout(0.2))

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1))

model.compile(optimizer=Adam(0.00001), loss='mse')

r = model.fit(X_train, y_train,
              validation_data=(X_test,y_test),
              batch_size=1,
              epochs=100)
```

```
Train on 3750 samples, validate on 1250 samples
Epoch 1/100
3750/3750 [==============================] - 9s 2ms/sample - loss:
1645011221832.2261 - val_loss: 1635384633501.4656
Epoch 2/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1644111549819.5627 - val_loss: 1633159231935.6160
Epoch 3/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1638650176479.2319 - val_loss: 1623126614041.7505
Epoch 4/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1620233921620.1045 - val_loss: 1594027084268.1597
Epoch 5/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1573934456872.4138 - val_loss: 1527222060114.8416
Epoch 6/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1478568384204.1003 - val_loss: 1399156169318.6047
Epoch 7/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1309945184051.1357 - val_loss: 1186604432128.3250
Epoch 8/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
1052174057382.0272 - val_loss: 888430125329.8519
Epoch 9/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
730868955302.6221 - val_loss: 565682944134.4716
```

```
Epoch 10/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
443362301239.5915 - val_loss: 346021754450.4661
Epoch 11/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
299598334345.9880 - val_loss: 275694459083.8212
Epoch 12/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
256707490892.1989 - val_loss: 251676776451.4203
Epoch 13/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
235936299151.4466 - val_loss: 234089969815.4346
Epoch 14/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
218921184426.8048 - val_loss: 218126958755.2028
Epoch 15/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
202971857334.3851 - val_loss: 203065281108.7159
Epoch 16/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
188627786744.3731 - val_loss: 188981114039.9154
Epoch 17/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
175087413838.6863 - val_loss: 175599196753.7520
Epoch 18/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
161834531938.6763 - val_loss: 163210906024.5396
Epoch 19/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
150567285921.6369 - val_loss: 151586497766.2021
Epoch 20/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
139585740740.3875 - val_loss: 140853718683.2468
Epoch 21/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
129219562361.1924 - val_loss: 130660716589.5126
Epoch 22/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
119487003620.8424 - val_loss: 121397918948.7557
Epoch 23/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
110820460034.8183 - val_loss: 112653657045.3199
Epoch 24/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
102589478501.4527 - val_loss: 104726786007.7076
Epoch 25/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
94478156359.9436 - val_loss: 97359566155.2182
```

```
Epoch 26/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
88192120617.3756 - val_loss: 90450376649.7863
Epoch 27/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
81487903978.5276 - val_loss: 84266620714.8715
Epoch 28/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
76252296633.4162 - val_loss: 78631932733.6203
Epoch 29/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
70865929461.1983 - val_loss: 73675350831.1541
Epoch 30/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
66231786207.5985 - val_loss: 69020007161.7298
Epoch 31/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
62053427538.0887 - val_loss: 64943240686.6493
Epoch 32/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
58527162565.4326 - val_loss: 61211352908.7847
Epoch 33/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
54685520842.1915 - val_loss: 57924572476.0930
Epoch 34/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
51788048028.5576 - val_loss: 54826443894.9727
Epoch 35/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
49141897299.4021 - val_loss: 52152889104.7891
Epoch 36/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
47030143373.8252 - val_loss: 49884710491.7800
Epoch 37/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
44828391032.6391 - val_loss: 47666203067.0596
Epoch 38/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
43517051412.2218 - val_loss: 45750516540.9983
Epoch 39/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
41354103922.9317 - val_loss: 44135337877.3657
Epoch 40/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
40387154127.5639 - val_loss: 42661729031.7819
Epoch 41/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
39004337165.8222 - val_loss: 41248256377.0208
```

```
Epoch 42/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
37871480944.8369 - val_loss: 40078781181.7467
Epoch 43/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
36912247927.6783 - val_loss: 38874481436.7543
Epoch 44/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
35881446417.8559 - val_loss: 37821308395.8920
Epoch 45/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
35016261587.8102 - val_loss: 36876120546.9743
Epoch 46/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
34175345612.7916 - val_loss: 36009781886.0239
Epoch 47/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
33297880993.2050 - val_loss: 35121148193.0421
Epoch 48/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
32606556160.4551 - val_loss: 34306464200.6651
Epoch 49/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
32025618355.6080 - val_loss: 33617151192.9297
Epoch 50/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
31051855755.5010 - val_loss: 32869033189.1942
Epoch 51/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
30610208864.4135 - val_loss: 32128095012.1506
Epoch 52/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
30001465354.1950 - val_loss: 31462279811.7975
Epoch 53/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
29555932365.0695 - val_loss: 30915481295.0542
Epoch 54/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
28632349903.3767 - val_loss: 30289330782.2692
Epoch 55/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
28622110292.0116 - val_loss: 29723876044.4669
Epoch 56/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
27950715939.2294 - val_loss: 29169539963.8105
Epoch 57/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
27433121418.1529 - val_loss: 28599910258.1587
```

```
Epoch 58/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
26917477115.8706 - val_loss: 28059722665.3292
Epoch 59/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
26439626483.7611 - val_loss: 27593364490.5045
Epoch 60/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
26136182428.1467 - val_loss: 27059995997.6096
Epoch 61/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
25579110730.6295 - val_loss: 26550143156.4183
Epoch 62/100
3750/3750 [==============================] - 4s 1ms/sample - loss:
25153242054.4432 - val_loss: 26243036570.4955
Epoch 63/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
24674340023.0935 - val_loss: 25784783137.4879
Epoch 64/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
24520607759.3302 - val_loss: 25250597357.8677
Epoch 65/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
24131140635.8410 - val_loss: 24886142117.1399
Epoch 66/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
23484612717.5341 - val_loss: 24519765866.2540
Epoch 67/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
23340953138.7429 - val_loss: 24060398297.5614
Epoch 68/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
22879659819.4470 - val_loss: 23690631303.1496
Epoch 69/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
22679756944.8186 - val_loss: 23322424558.2063
Epoch 70/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
22228570482.0951 - val_loss: 22972429783.7407
Epoch 71/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
22071594373.7687 - val_loss: 22681682113.0295
Epoch 72/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
21443544090.9248 - val_loss: 22335733000.9896
Epoch 73/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
21438727055.3090 - val_loss: 22027554742.1772
```

```
Epoch 74/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
21099365454.3829 - val_loss: 21698968507.3483
Epoch 75/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
20904163976.7863 - val_loss: 21437286991.3694
Epoch 76/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
20409710610.1618 - val_loss: 21167442120.6133
Epoch 77/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
20496735916.6194 - val_loss: 20922356915.3711
Epoch 78/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
20109176423.3906 - val_loss: 20648921624.0706
Epoch 79/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
19741266142.2567 - val_loss: 20378445395.6966
Epoch 80/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
19649727429.8518 - val_loss: 20140318040.4399
Epoch 81/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
19626618292.1871 - val_loss: 19890450867.4310
Epoch 82/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
19316079378.7028 - val_loss: 19658271751.2137
Epoch 83/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18931874991.2881 - val_loss: 19409373402.4451
Epoch 84/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18960815463.4678 - val_loss: 19181496279.5640
Epoch 85/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18484176559.3801 - val_loss: 18956720865.4312
Epoch 86/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18585703306.4921 - val_loss: 18748773853.8683
Epoch 87/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18294984430.8272 - val_loss: 18594847227.8874
Epoch 88/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18128920476.2852 - val_loss: 18415953266.3600
Epoch 89/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
18097334044.5695 - val_loss: 18215937396.1826
```

```
Epoch 90/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
17934898477.6272 - val_loss: 18124597981.6971
Epoch 91/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
17726429623.2591 - val_loss: 17939115067.6198
Epoch 92/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
17582098581.8906 - val_loss: 17730811276.8088
Epoch 93/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
17088822595.4930 - val_loss: 17564127314.6064
Epoch 94/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
17025575506.3072 - val_loss: 17351329460.5691
Epoch 95/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16992228691.3439 - val_loss: 17248361139.7766
Epoch 96/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16906260257.4238 - val_loss: 17100072629.8632
Epoch 97/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16994846568.3757 - val_loss: 17009352127.3486
Epoch 98/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16873500050.7127 - val_loss: 16841243958.8149
Epoch 99/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16665831336.8889 - val_loss: 16639723686.5436
Epoch 100/100
3750/3750 [==============================] - 5s 1ms/sample - loss:
16402260027.3531 - val_loss: 16500752288.3134
```

[75]:
```
pd.DataFrame(r.history)
```

[75]:

	loss	val_loss
0	1.645011e+12	1.635385e+12
1	1.644112e+12	1.633159e+12
2	1.638650e+12	1.623127e+12
3	1.620234e+12	1.594027e+12
4	1.573934e+12	1.527222e+12
..	…	…
95	1.690626e+10	1.710007e+10
96	1.699485e+10	1.700935e+10
97	1.687350e+10	1.684124e+10
98	1.666583e+10	1.663972e+10

```
99  1.640226e+10  1.650075e+10

[100 rows x 2 columns]
```

[76]:

```
pd.DataFrame(r.history).hvplot.line(y=['loss', 'val_loss'])
```

[76]:

```
:NdOverlay   [Variable]
   :Curve   [index]   (value)
```