```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

# a) Read the data with pandas and describe the data
data = pd.read_csv("/content/housing.csv")
data = data.drop('ocean_proximity',axis=1)
print("Description of the data:")
print(data.describe())

# b) Find data type and shape of each column
print("\nData types of each column:")
print(data.dtypes)
print("\nShape of the data:")
print(data.shape)

# c) Find the null values and fill them with mean of that column
null_values = data.isnull().sum()
print("\nNull values in the data:")
print(null_values)

# Fill null values with mean
data.fillna(data.mean(), inplace=True)

# d) Find features and target variables
features = data.drop(columns=['median_house_value'])
target = data['median_house_value']
print(features)
print(target)

# e) Split the data into train and test
X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.2, random_state=42)

# f) Normalize the data with min-max scaling
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
print("\nThe Normalized Data:\n")
print("Train Scaled\n")
print(X_train_scaled)
print("\nTest Scaled\n")
print(X_test_scaled)
```

```
Description of the data:
          longitude      latitude  housing_median_age   total_rooms  \
count  20640.000000  20640.000000        20640.000000  20640.000000
mean    -119.569704     35.631861           28.639486   2635.763081
std        2.003532      2.135952           12.585558   2181.615252
min     -124.350000     32.540000            1.000000      2.000000
```

```
25%     -121.800000     33.930000           18.000000   1447.750000
50%     -118.490000     34.260000           29.000000   2127.000000
75%     -118.010000     37.710000           37.000000   3148.000000
max     -114.310000     41.950000           52.000000  39320.000000

       total_bedrooms    population    households  median_income  \
count    20433.000000  20640.000000  20640.000000   20640.000000
mean       537.870553   1425.476744    499.539680       3.870671
std        421.385070   1132.462122    382.329753       1.899822
min          1.000000      3.000000      1.000000       0.499900
25%        296.000000    787.000000    280.000000       2.563400
50%        435.000000   1166.000000    409.000000       3.534800
75%        647.000000   1725.000000    605.000000       4.743250
max       6445.000000  35682.000000   6082.000000      15.000100

       median_house_value
count        20640.000000
mean        206855.816909
std         115395.615874
min          14999.000000
25%         119600.000000
50%         179700.000000
75%         264725.000000
max         500001.000000

Data types of each column:
longitude             float64
latitude              float64
housing_median_age    float64
total_rooms           float64
total_bedrooms        float64
population            float64
households            float64
median_income         float64
median_house_value    float64
dtype: object

Shape of the data:
(20640, 9)

Null values in the data:
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
```

```
dtype: int64
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
0        -122.23     37.88                41.0        880.0
129.0
1        -122.22     37.86                21.0       7099.0
1106.0
2        -122.24     37.85                52.0       1467.0
190.0
3        -122.25     37.85                52.0       1274.0
235.0
4        -122.25     37.85                52.0       1627.0
280.0
...          ...       ...                 ...          ...
...
20635    -121.09     39.48                25.0       1665.0
374.0
20636    -121.21     39.49                18.0        697.0
150.0
20637    -121.22     39.43                17.0       2254.0
485.0
20638    -121.32     39.43                18.0       1860.0
409.0
20639    -121.24     39.37                16.0       2785.0
616.0

       population  households  median_income
0           322.0       126.0         8.3252
1          2401.0      1138.0         8.3014
2           496.0       177.0         7.2574
3           558.0       219.0         5.6431
4           565.0       259.0         3.8462
...           ...         ...            ...
20635       845.0       330.0         1.5603
20636       356.0       114.0         2.5568
20637      1007.0       433.0         1.7000
20638       741.0       349.0         1.8672
20639      1387.0       530.0         2.3886

[20640 rows x 8 columns]
0        452600.0
1        358500.0
2        352100.0
3        341300.0
4        342200.0
           ...
20635     78100.0
20636     77100.0
20637     92300.0
```

```
20638     84700.0
20639     89400.0
Name: median_house_value, Length: 20640, dtype: float64

The Normalized Data:

Train Scaled

[[0.72908367 0.01702128 0.62745098 ... 0.06437961 0.10228581
0.19032151]
 [0.61653386 0.12978723 0.94117647 ... 0.0367443  0.12415721
0.22845202]
 [0.38545817 0.22446809 0.05882353 ... 0.02556125 0.05508962
0.25216204]
 ...
 [0.59462151 0.15744681 0.68627451 ... 0.04913254 0.08649893
0.16789424]
 [0.23804781 0.53510638 0.2745098  ... 0.04972112 0.09176122
0.35994676]
 [0.19223108 0.55531915 1.         ... 0.07332044 0.20407828
0.14314285]]

Test Scaled

[[0.53187251 0.37340426 0.47058824 ... 0.03893046 0.0588719
0.08146784]
 [0.48705179 0.27553191 0.56862745 ... 0.04377925 0.09587239
0.14009462]
 [0.19023904 0.55851064 1.         ... 0.03663219 0.15819766 0.2055282
]
 ...
 [0.22908367 0.50638298 0.47058824 ... 0.04433981 0.09324124
0.60205376]
 [0.45717131 0.44893617 0.68627451 ... 0.03430589 0.07778326
0.15759093]
 [0.59561753 0.17765957 0.31372549 ... 0.04675019 0.07350765
0.21049365]]
```