ML Internship Task - 01

Implement a linear regression model to predict the prices of houses based on their
square footage and the number of bedrooms and bathrooms.

importing libraries like pandas and dataset

In [1]:
```
import pandas as pd
```

In [2]:

```
df=pd.read_csv("house_prediction.csv")
print(df)
```

```
                  ADDRESS           SUBURB    PRICE  BEDROOMS  BATHROO
MS  \
0           1 Acorn Place       South Lake   565000         4
2
1             1 Addis Way            Wandi   365000         3
2
2         1 Ainsley Court          Camillo   287000         3
1
3         1 Albert Street         Bellevue   255000         2
1
4            1 Aman Place        Lockridge   325000         4
1
...                   ...              ...      ...       ...
...
33651      9C Gold Street  South Fremantle  1040000         4
3
33652      9C Pycombe Way      Westminster   410000         3
2
33653      9D Pycombe Way      Westminster   427000         3
2
33654     9D Shalford Way       Girrawheen   295000         3
1
33655  9E Margaret Street          Midland   295000         3
1

       GARAGE  LAND_AREA  FLOOR_AREA  BUILD_YEAR  CBD_DIST  \
0         2.0        600         160      2003.0     18300
1         2.0        351         139      2013.0     26900
2         1.0        719          86      1979.0     22600
3         2.0        651          59      1953.0     17900
4         2.0        466         131      1998.0     11200
...       ...        ...         ...         ...       ...
33651     2.0        292         245      2013.0     16100
33652     2.0        228         114         NaN      9600
33653     2.0        261         112         NaN      9600
33654     2.0        457          85      1974.0     12600
33655     2.0        296          95         NaN     16700

                   NEAREST_STN  NEAREST_STN_DIST  DATE_SOLD  POSTCODE
\
0      Cockburn Central Station              1800  09-2018\r      6164
1               Kwinana Station              4900  02-2019\r      6167
2               Challis Station              1900  06-2015\r      6111
3               Midland Station              3600  07-2018\r      6056
4            Bassendean Station              2000  11-2016\r      6054
...                         ...               ...        ...       ...
33651         Fremantle Station              1500  03-2016\r      6162
33652          Stirling Station              4600  02-2017\r      6061
33653          Stirling Station              4600  02-2017\r      6061
33654           Warwick Station              4400  10-2016\r      6064
33655           Midland Station              1700  05-2016\r      6056

        LATITUDE   LONGITUDE                                NEAREST_
SCH  \
0     -32.115900  115.842450                LAKELAND SENIOR HIGH SCH
OOL
1     -32.193470  115.859554                             ATWELL COLL
EGE
2     -32.120578  115.993579               KELMSCOTT SENIOR HIGH SCH
OOL
3     -31.900547  116.038009               SWAN VIEW SENIOR HIGH SCH
```

```
OOL
4     -31.885790  115.947780                                KIARA COLL
EGE
...          ...         ...                                       
...
33651 -32.064580  115.751820                 CHRISTIAN BROTHERS' COLL
EGE
33652 -31.867055  115.841403  JOHN SEPTIMUS ROE ANGLICAN COMMUNITY SCH
OOL
33653 -31.866890  115.841418  JOHN SEPTIMUS ROE ANGLICAN COMMUNITY SCH
OOL
33654 -31.839680  115.842410               GIRRAWHEEN SENIOR HIGH SCH
OOL
33655 -31.882163  116.014755                           LA SALLE COLL
EGE

       NEAREST_SCH_DIST  NEAREST_SCH_RANK
0              0.828339               NaN
1              5.524324             129.0
2              1.649178             113.0
3              1.571401               NaN
4              1.514922               NaN
...                 ...               ...
33651          1.430350              49.0
33652          1.679644              35.0
33653          1.669159              35.0
33654          0.358494               NaN
33655          1.055564              53.0

[33656 rows x 19 columns]
```

In [3]:
```
df.head()
```

Out[3]:

	ADDRESS	SUBURB	PRICE	BEDROOMS	BATHROOMS	GARAGE	LAND_AREA	FL
0	1 Acorn Place	South Lake	565000	4	2	2.0	600	
1	1 Addis Way	Wandi	365000	3	2	2.0	351	
2	1 Ainsley Court	Camillo	287000	3	1	1.0	719	
3	1 Albert Street	Bellevue	255000	2	1	2.0	651	
4	1 Aman Place	Lockridge	325000	4	1	2.0	466	

In [5]:
```
df.tail()
```

Out[5]:

	ADDRESS	SUBURB	PRICE	BEDROOMS	BATHROOMS	GARAGE	LAND_AI
33651	9C Gold Street	South Fremantle	1040000	4	3	2.0	
33652	9C Pycombe Way	Westminster	410000	3	2	2.0	
33653	9D Pycombe Way	Westminster	427000	3	2	2.0	
33654	9D Shalford Way	Girrawheen	295000	3	1	2.0	
33655	9E Margaret Street	Midland	295000	3	1	2.0	

In [6]:
```
df.describe()
```

Out[6]:

	PRICE	BEDROOMS	BATHROOMS	GARAGE	LAND_AREA	FLOOR_
count	3.365600e+04	33656.000000	33656.000000	31178.000000	33656.000000	33656.0
mean	6.370720e+05	3.659110	1.823063	2.199917	2740.644016	183.5
std	3.558256e+05	0.752038	0.587427	1.365225	16693.513215	72.1
min	5.100000e+04	1.000000	1.000000	1.000000	61.000000	1.0
25%	4.100000e+05	3.000000	1.000000	2.000000	503.000000	130.0
50%	5.355000e+05	4.000000	2.000000	2.000000	682.000000	172.0
75%	7.600000e+05	4.000000	2.000000	2.000000	838.000000	222.2
max	2.440000e+06	10.000000	16.000000	99.000000	999999.000000	870.0

In [7]:
```
df.columns
```

Out[7]:
```
Index(['ADDRESS', 'SUBURB', 'PRICE', 'BEDROOMS', 'BATHROOMS', 'GARAG
E',
       'LAND_AREA', 'FLOOR_AREA', 'BUILD_YEAR', 'CBD_DIST', 'NEAREST_S
TN',
       'NEAREST_STN_DIST', 'DATE_SOLD', 'POSTCODE', 'LATITUDE', 'LONGI
TUDE',
       'NEAREST_SCH', 'NEAREST_SCH_DIST', 'NEAREST_SCH_RANK'],
      dtype='object')
```

In [8]:
```
df.dtypes
```

Out[8]:
```
ADDRESS              object
SUBURB               object
PRICE                 int64
BEDROOMS              int64
BATHROOMS             int64
GARAGE              float64
LAND_AREA             int64
FLOOR_AREA            int64
BUILD_YEAR          float64
CBD_DIST              int64
NEAREST_STN          object
NEAREST_STN_DIST      int64
DATE_SOLD            object
POSTCODE              int64
LATITUDE            float64
LONGITUDE           float64
NEAREST_SCH          object
NEAREST_SCH_DIST    float64
NEAREST_SCH_RANK    float64
dtype: object
```

In [26]:
```
df.isnull().sum()
```

Out[26]:
```
ADDRESS                0
SUBURB                 0
PRICE                  0
BEDROOMS               0
BATHROOMS              0
GARAGE              2478
LAND_AREA              0
FLOOR_AREA             0
BUILD_YEAR          3155
CBD_DIST               0
NEAREST_STN            0
NEAREST_STN_DIST       0
DATE_SOLD              0
POSTCODE               0
LATITUDE               0
LONGITUDE              0
NEAREST_SCH            0
NEAREST_SCH_DIST       0
NEAREST_SCH_RANK    10952
dtype: int64
```

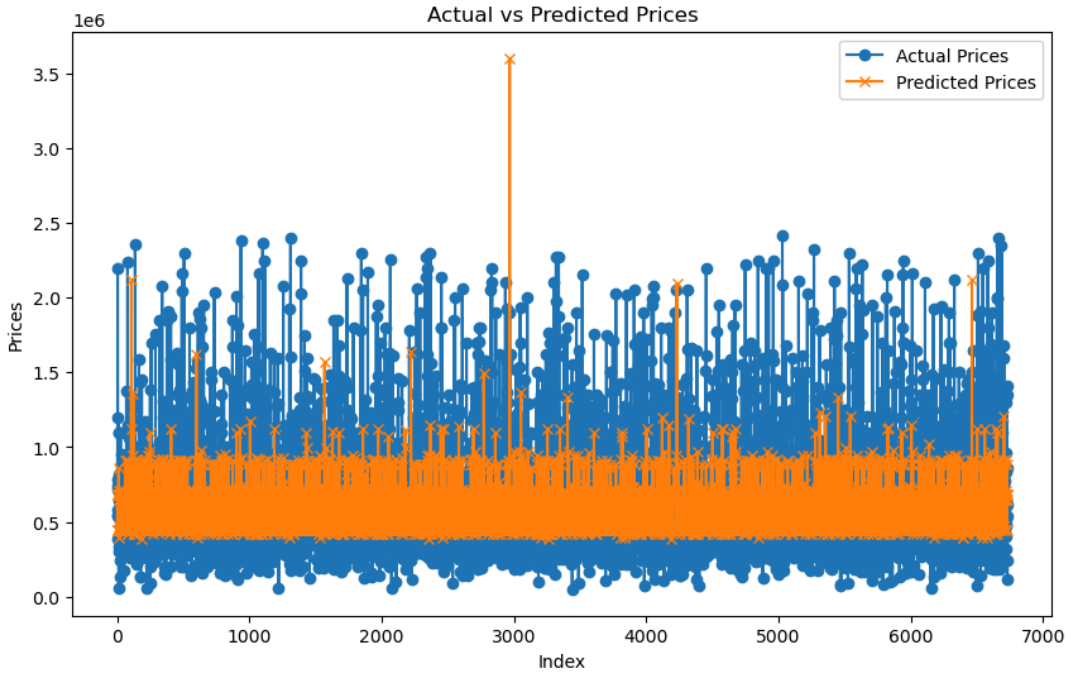
In [32]:
```
# Fill missing values with the mean of the respective column
df.fillna(df.mean(), inplace=True)

# Verify that there are no missing values
print(df.isnull().sum())
```

```
ADDRESS             0
SUBURB              0
PRICE               0
BEDROOMS            0
BATHROOMS           0
GARAGE              0
LAND_AREA           0
FLOOR_AREA          0
BUILD_YEAR          0
CBD_DIST            0
NEAREST_STN         0
NEAREST_STN_DIST    0
DATE_SOLD           0
POSTCODE            0
LATITUDE            0
LONGITUDE           0
NEAREST_SCH         0
NEAREST_SCH_DIST    0
NEAREST_SCH_RANK    0
dtype: int64

C:\Users\senap\AppData\Local\Temp\ipykernel_7264\3294738352.py:2: Futu
reWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will rais
e TypeError.  Select only valid columns before calling the reduction.
  df.fillna(df.mean(), inplace=True)
```

In [38]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Split the data into features and target variable
X = df[['LAND_AREA','BEDROOMS','BATHROOMS']]
y = df['PRICE']

# 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

# Build the linear regression model
model = LinearRegression()

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

# Make predictions
y_pred = model.predict(X_test)

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

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Plot the results
plt.figure(figsize=(10, 6))
plt.plot(y_test.values, label='Actual Prices', marker='o')
plt.plot(y_pred, label='Predicted Prices', marker='x')
plt.xlabel("Index")
plt.ylabel("Prices")
plt.title("Actual vs Predicted Prices")
plt.legend()
plt.show()
```

```
Mean Squared Error: 112746942404.97926
R-squared: 0.1402771529472947
```


Actual vs Predicted Prices
1e6
Actual Prices
Predicted Prices
Prices
Index
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
1000
2000
3000
4000
5000
6000
7000

In [ ]:

In [ ]:

In [ ]:

In [ ]: