In [3]:
```
import numpy as np
import pandas as pd   # importing to calculate numeric operation and for dat
```

In [5]:
```
df = pd.read_csv('/Users/sagarbanjara/Desktop/Takeo/AmesHousing.csv')    # i
```

In [7]:
```
missing_values = df.isnull().sum()  #finding the missing values
```

In [9]:
```
# df = df.select_dtypes(exclude = 'bool')
df= df.select_dtypes(include=['int64', 'float64'])   #there are some boolean
```

In [11]:
```
df.drop_duplicates(inplace = True)   # deleting the duplicate rows
```

In [13]:
```
df= df.dropna(axis=1, thresh=int(0.8 * len(df)))   #calculating the threshol
```

In [15]:
```
df.fillna(df.median(), inplace=True)      #filling missing values
```

In [17]:
```
df = pd.get_dummies(df, drop_first = True)     #converting category into num
```

In [19]:
```
x = df.drop("SalePrice", axis = 1)      # defining the value for x and y
y = df["SalePrice"]
```

In [21]:
```
from sklearn.model_selection import train_test_split      #importing the nec
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [22]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, ran
model = LinearRegression()         ##model we are supposed to use
model.fit(x_train,y_train)      #connecting relationship
```

Out[22]:
```
▾ LinearRegression

LinearRegression()
```

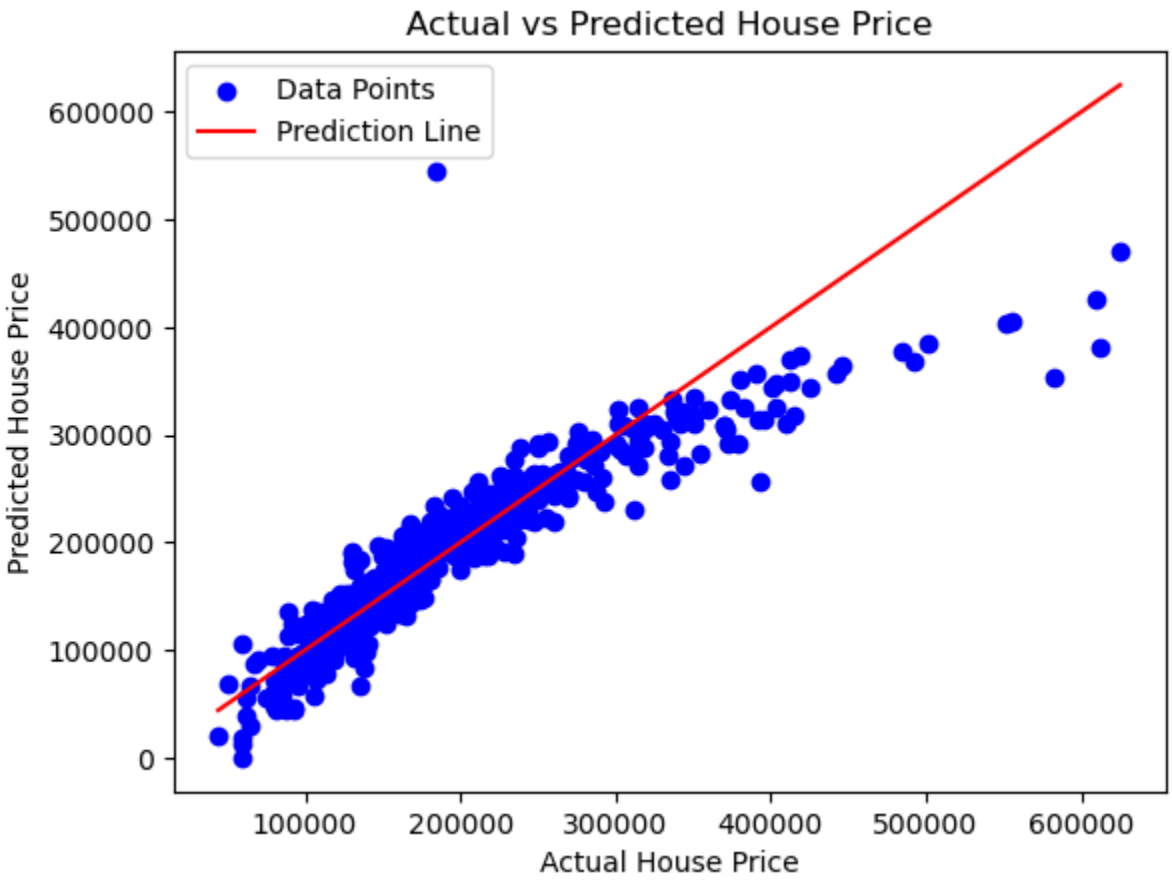
In [23]:
```
print(x.dtypes.value_counts())       #checking the data types
```

```
int64      27
float64    11
Name: count, dtype: int64
```

In [24]:
```
predicting_house_price = model.predict(x_test)     #how to predict the house
```

In [55]:
```
slope = model.coef_[0]           # shows the
intercept = model.intercept_
```

In [57]:
```
plt.scatter(y_test, predicting_house_price, color='blue', label='Data Points
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='
plt.xlabel('Actual House Price')
plt.ylabel('Predicted House Price')
plt.title('Actual vs Predicted House Price')        #giving the specific nam
plt.legend()
plt.show()
```



In [ ]: