2.5. Predicting Apartment Prices in Mexico City 🇲🇽

In [30]:

```
import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)
```

In this assignment, you'll decide which libraries you need to complete the tasks. You can import them in the cell below. 👇

In [32]:

```
# Import libraries here
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from glob import glob
import plotly.express as px
from sklearn.metrics import mean_absolute_error
from sklearn.pipeline import make_pipeline
from sklearn.linear_model import Ridge
from category_encoders import OneHotEncoder
from sklearn.impute import SimpleImputer
```

Prepare Data

Import

Task 2.5.1

Tip: Don't try to satisfy all the criteria in the first version of your `wrangle` function. Instead, work iteratively. Start with the first criteria, test it out with one of the Mexico CSV files in the `data/` directory, and submit it to the grader for feedback. Then add the next criteria.

In [3]:

```
# Build your `wrangle` function
def wrangle(path):
    df = pd.read_csv(path)
    mask_pw = df['place_with_parent_names'].str.contains("Distrito Federa
    mask_pa = df['property_type'] == 'apartment'
    mask_price = df['price_aprox_usd'] < 100000
    df = df[mask_pw & mask_pa & mask_price]

    low, high = df["surface_covered_in_m2"].quantile([0.1, 0.9])
    df = df[df['surface_covered_in_m2'].between(low, high)]

    df[['lat','lon']] = df['lat-lon'].str.split(",", expand=True).astype(
    df['borough'] = df['place_with_parent_names'].str.split('|', expand=T

    df.drop(columns=[
        'lat-lon','surface_total_in_m2','price_usd_per_m2','floor','rooms
        'expenses','place_with_parent_names','operation','property_type',
        'price','currency','price_aprox_local_currency','price_per_m2',
        'properati_url'
    ], inplace=True)

    return df
```

In [4]:

```
# Use this cell to test your wrangle function on the file `mexico-city-re
frame1 = wrangle('data/mexico-city-real-estate-1.csv')
frame1.head(10)
```

Out[4]:

	price_aprox_usd	surface_covered_in_m2	lat	lon	borough
11	94022.66	57.0	23.634501	-102.552788	Benito Juárez
20	70880.12	56.0	19.402413	-99.095391	Iztacalco
21	68228.99	80.0	19.357820	-99.149406	Benito Juárez
22	24235.78	60.0	19.504985	-99.208557	Azcapotzalco
26	94140.20	50.0	19.354219	-99.126244	Coyoacán
32	76395.40	90.0	19.400785	-99.147419	Benito Juárez
43	40103.52	65.0	19.366632	-99.197900	Álvaro Obregón
46	68070.93	80.0	19.400731	-99.125466	Iztacalco
51	39409.48	61.0	19.337163	-99.080899	Iztapalapa
59	44794.13	55.0	19.435666	-99.149730	Cuauhtémoc

In [5]:

```
frame1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1101 entries, 11 to 4605
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   price_aprox_usd       1101 non-null   float64
 1   surface_covered_in_m2 1101 non-null   float64
 2   lat                   1041 non-null   float64
 3   lon                   1041 non-null   float64
 4   borough               1101 non-null   object 
dtypes: float64(4), object(1)
memory usage: 51.6+ KB
```

Task 2.5.2

In [6]:

```
files = glob('data/mexico-city-real-estate-*.csv')
files
```

Out[6]:

```
['data/mexico-city-real-estate-1.csv',
 'data/mexico-city-real-estate-2.csv',
 'data/mexico-city-real-estate-3.csv',
 'data/mexico-city-real-estate-4.csv',
 'data/mexico-city-real-estate-5.csv']
```

Task 2.5.3

In [7]:

```
frames =[]
for file in files:
    d = wrangle(file)
    frames.append(d)
```

In [8]:

```
df = pd.concat(frames,ignore_index = True)
print(df.info())
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5473 entries, 0 to 5472
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   price_aprox_usd      5473 non-null   float64
 1   surface_covered_in_m2  5473 non-null   float64
 2   lat                  5149 non-null   float64
 3   lon                  5149 non-null   float64
 4   borough              5473 non-null   object 
dtypes: float64(4), object(1)
memory usage: 213.9+ KB
None
```

Out[8]:

	price_aprox_usd	surface_covered_in_m2	lat	lon	borough
0	94022.66	57.0	23.634501	-102.552788	Benito Juárez
1	70880.12	56.0	19.402413	-99.095391	Iztacalco
2	68228.99	80.0	19.357820	-99.149406	Benito Juárez
3	24235.78	60.0	19.504985	-99.208557	Azcapotzalco
4	94140.20	50.0	19.354219	-99.126244	Coyoacán

Explore

Slight Code Change

In the following task, you'll notice a small change in how plots are created compared to what you saw in the lessons. While the lessons use the global matplotlib method like `plt.plot(...)`, in this task, you are expected to use the object-oriented (OOP) API instead. This means creating your plots using `fig, ax = plt.subplots()` and then calling plotting methods on the `ax` object, such as `ax.plot(...)`, `ax.hist(...)`, or `ax.scatter(...)`.

If you're using pandas' or seaborn's built-in plotting methods (like `df.plot()` or `sns.lineplot()`), make sure to pass the `ax=ax` argument so that the plot is rendered on the correct axes.

This approach is considered best practice and will be used consistently across all graded tasks that involve matplotlib.

Task 2.5.4

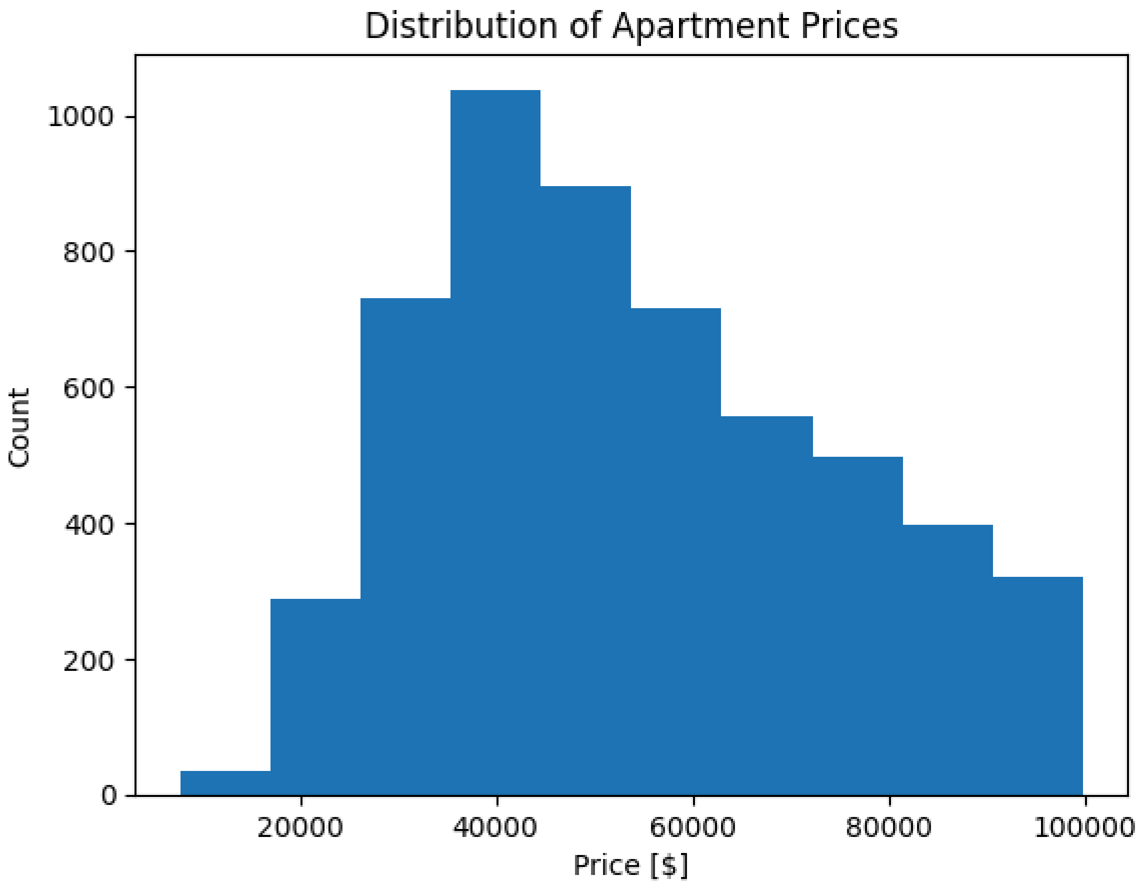
In [9]:

```
fig, ax = plt.subplots()

# Plot the histogram on the axes object
ax.hist(df['price_aprox_usd'])

# Label axes using the axes
ax.set_xlabel("Price [$]")
ax.set_ylabel("Count")


# Add title
ax.set_title("Distribution of Apartment Prices")
plt.show()
```



Task 2.5.5

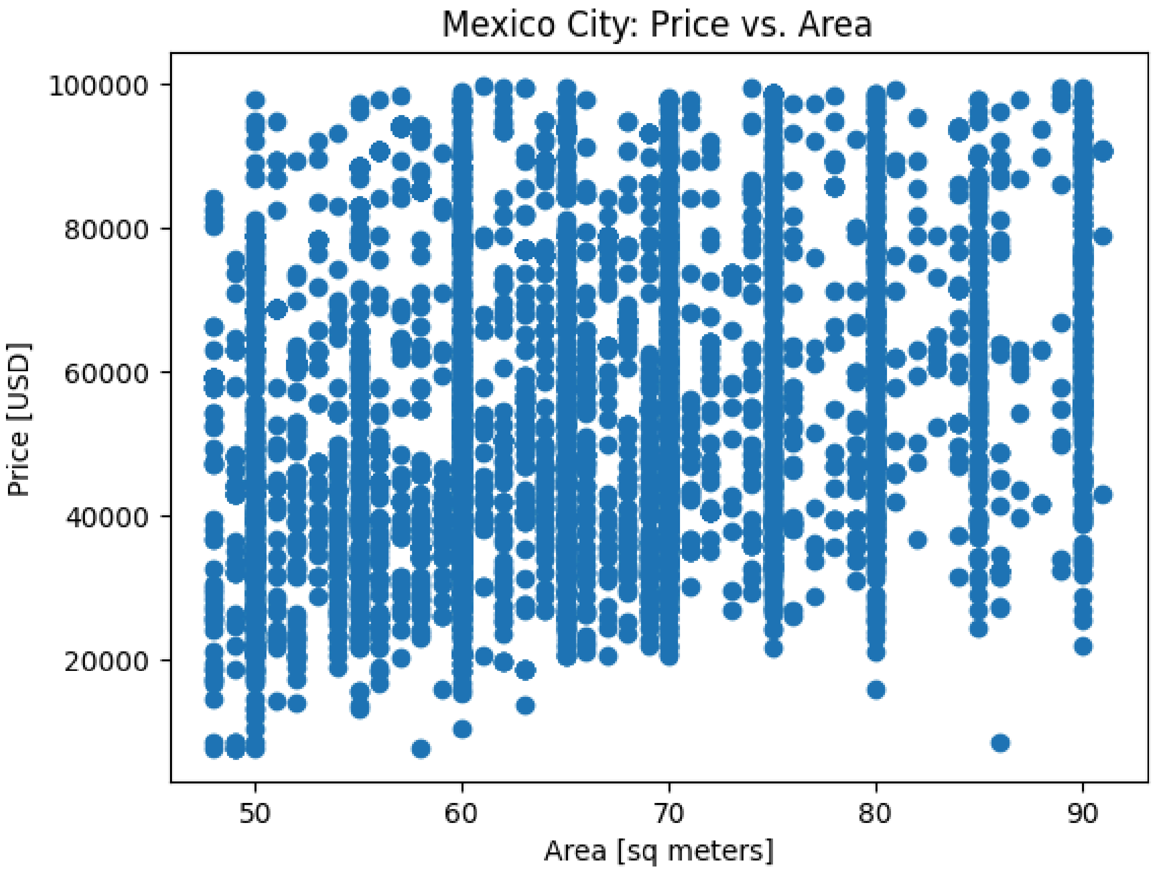
In [10]:

```
fig, ax = plt.subplots()

# Create the scatter plot on the axes object
ax.scatter(x=df['surface_covered_in_m2'],y=df["price_aprox_usd"])
```

```
# Label axes
ax.set_xlabel("Area [sq meters]")
ax.set_ylabel("Price [USD]")

#  Add title
ax.set_title("Mexico City: Price vs. Area")
plt.show()
```



Do you see a relationship between price and area in the data? How is this similar to or different from the Buenos Aires dataset?

Task 2.5.6 (UNGRADED) Create a Mapbox scatter plot that shows the location of the apartments in your dataset and represent their price using color.

What areas of the city seem to have higher real estate prices?

In [18]:
```
# Plot Mapbox location and price

fig = px.scatter_mapbox(
    df,
    lat='lat',
    lon= 'lon',
    width=600,
    height=600,
    color= 'price_aprox_usd',
    hover_data=["price_aprox_usd"],
)

fig.update_layout(mapbox_style="open-street-map")
fig.show()
```

Split

Task 2.5.7

In [19]:

```
# Split data into feature matrix `X_train` and target vector `y_train`.
features = ['surface_covered_in_m2','lat','lon','borough']
X_train = df[features]
y_train = df['price_aprox_usd']
```

Build Model

Baseline

Task 2.5.8

In [24]:

```
y_mean = y_train.mean()
y_pred_baseline = len(y_train)*[y_mean]
baseline_mae = mean_absolute_error(y_train,y_pred_baseline)
print("Mean apt price:", y_mean)
print("Baseline MAE:", baseline_mae)
```

```
Mean apt price: 54246.5314982642
Baseline MAE: 17239.939475888295
```

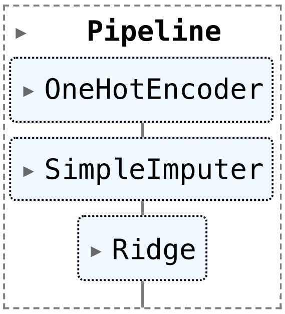
Iterate

Task 2.5.9

In [34]:

```
# Build Model
model = make_pipeline(
    OneHotEncoder(use_cat_names=True),
    SimpleImputer(),
    Ridge()
)
# Fit model
model.fit(X_train,y_train)
```

Out[34]:



▸ Pipeline

▸ OneHotEncoder

▸ SimpleImputer

▸ Ridge

Evaluate

Task 2.5.10

Tip: Make sure the `X_train` you used to train your model has the same column order as `X_test`. Otherwise, it may hurt your model's performance.

In [37]:

```
X_test = pd.read_csv('data/mexico-city-test-features.csv')
print(X_test.info())
X_test.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1041 entries, 0 to 1040
Data columns (total 4 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   surface_covered_in_m2  1041 non-null   float64
 1   lat                  986 non-null    float64
 2   lon                  986 non-null    float64
 3   borough              1041 non-null   object 
dtypes: float64(3), object(1)
memory usage: 32.7+ KB
None
```

Out[37]:

	surface_covered_in_m2	lat	lon	borough
0	60.0	19.493185	-99.205755	Azcapotzalco
1	55.0	19.307247	-99.166700	Coyoacán
2	50.0	19.363469	-99.010141	Iztapalapa
3	60.0	19.474655	-99.189277	Azcapotzalco
4	74.0	19.394628	-99.143842	Benito Juárez

Task 2.5.11

In [40]:

```
y_test_pred = pd.Series(model.predict(X_test))
y_test_pred.head()
```

Out[40]:

```
0    53538.366480
1    53171.988369
2    34263.884179
3    53488.425607
4    68738.924884
dtype: float64
```

Communicate Results

Task 2.5.12

In [50]:

```
coefficients = model.named_steps['ridge'].coef_
features = model.named_steps['onehotencoder'].get_feature_names()
feat_imp = pd.Series(coefficients,index=features)
feat_imp = feat_imp.reindex(feat_imp.abs().sort_values(ascending=True).in
feat_imp
```

Out[50]:

```
surface_covered_in_m2                291.654156
borough_Cuauhtémoc                  -350.531990
borough_Iztacalco                    405.403127
lat                                  478.901375
borough_Xochimilco                   929.857400
borough_Miguel Hidalgo              1977.314718
borough_Azcapotzalco                2459.288646
lon                                -2492.221814
borough_Álvaro Obregón              3275.121061
borough_Coyoacán                    3737.561001
borough_Venustiano Carranza        -5609.918629
borough_La Magdalena Contreras     -5925.666450
borough_Gustavo A. Madero          -6637.429757
borough_Cuajimalpa de Morelos       9157.269123
borough_Tlalpan                    10319.429804
borough_Iztapalapa                -13349.017448
borough_Benito Juárez              13778.188880
borough_Tláhuac                   -14166.869486
dtype: float64
```

Task 2.5.13

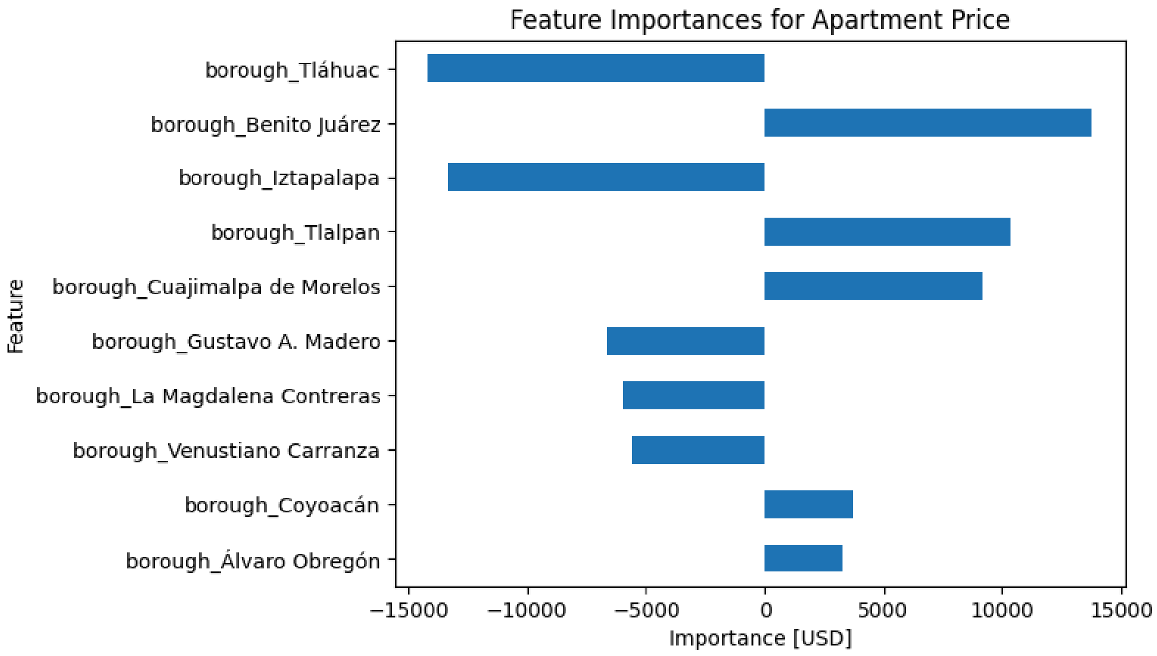
In [56]:

```python
fig, ax = plt.subplots()

# Create the horizontal bar plot on the axes object
feat_imp.tail(10).plot(kind='barh', ax=ax)

#  Label axes
ax.set_xlabel("Importance [USD]")
ax.set_ylabel("Feature")

# Add title
ax.set_title("Feature Importances for Apartment Price")
plt.show()
```


Feature Importances for Apartment Price
borough_Tláhuac
borough_Benito Juárez
borough_Iztapalapa
borough_Tlalpan
borough_Cuajimalpa de Morelos
borough_Gustavo A. Madero
borough_La Magdalena Contreras
borough_Venustiano Carranza
borough_Coyoacán
borough_Álvaro Obregón
Feature
Importance [USD]