King County Housing with Multiple Linear Regression

Authors: Diane Tunnicliffe, Dana Rausch, Matthew Lipman

Notebook 4: Visualizations

This notebook contains visualizations to illustrate our findings from our analysis and our key features as designated by our final model.

In [43]:
```
import numpy as np
import folium
from folium import plugins
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import branca.colormap as cm
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [44]:
```
df = pd.read_csv('./data/all_features_ppsqft_quant.csv', index_col=0)
top_schools_df = pd.read_csv('data/top_schools.csv')
df_orig = pd.read_csv('data/kc_house_data.csv')
```

In [45]:
```
def get_center_latlong(df):
    # get the center of my map for plotting
    centerlat = (df['lat'].max() + df['lat'].min()) / 2
    centerlong = (df['long'].max() + df['long'].min()) / 2
    return centerlat, centerlong
```

In [46]:
```
# grab the center of our map using our cool function
center = get_center_latlong(df)

# create a new map object
le_map = folium.Map(location=center, zoom_start=9)
```

Mapping Price

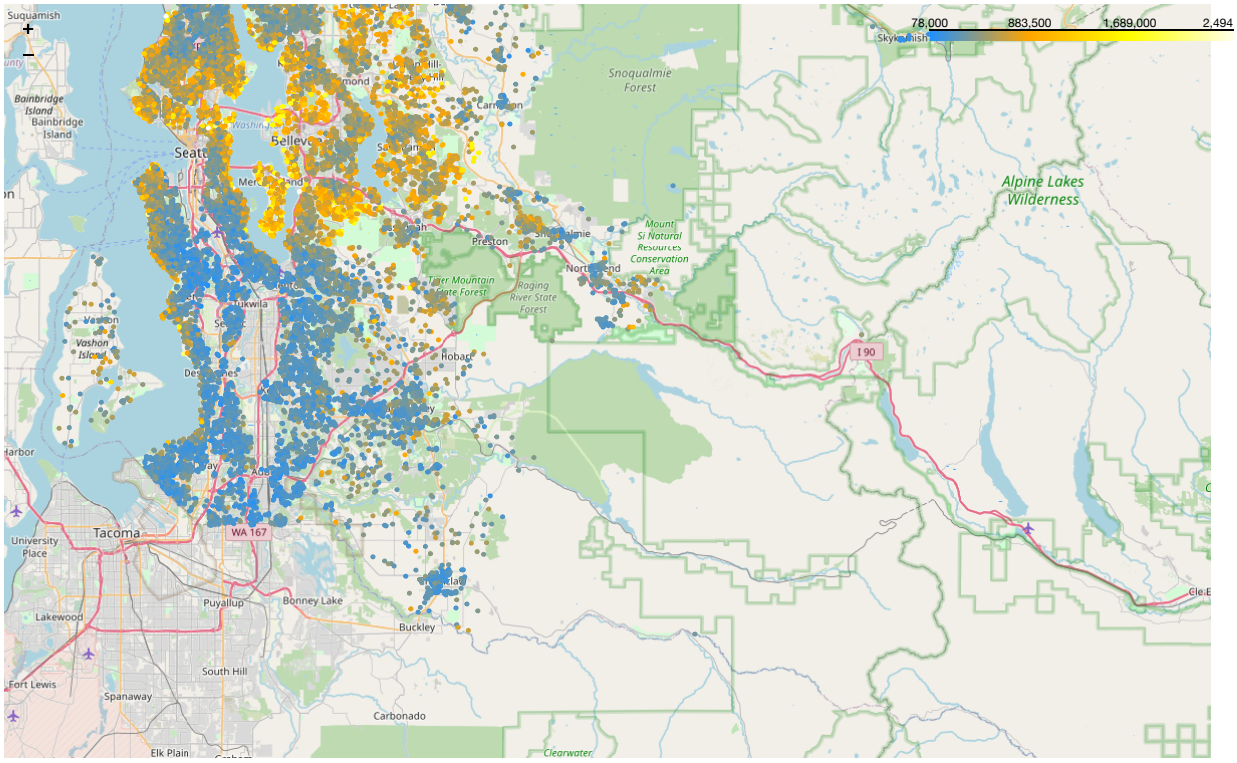
In [47]:
```
# create a LinearColorMap and assign color gradient -> PRICE
colormap = cm.LinearColormap(colors=['dodgerblue', 'orange', 'yellow', 'lightyellow', 'snow'], vmin=df['price'].min(), vmax=df['price'].max())

# create our map again.  This time using a different tileset for a new look
le_map = folium.Map(location=center, zoom_start=10)

# Same as before... go through each home in set, make circle, and add to map.
# This time we add a color using price and the colormap object
for i in range(len(df)):
    folium.Circle(
        location=[df.iloc[i]['lat'], df.iloc[i]['long']],
        radius=10,
        fill=True,
        color=colormap(df.iloc[i]['price']),
        fill_opacity=0.2
    ).add_to(le_map)

# the following line adds the scale directly to our map
le_map.add_child(colormap)
```

Out[47]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.open

In [48]:
```
le_map.save('./visualizations/price_map.png')
```

This map displays the geographical distribution of house prices in King County. We can see that the majority of them are blue, which is on the lower end of the price range shown in the data. Howev given range, these are actually the prices that are considered average for this sample in terms of mean, median, and mode.

Sqft Living | R-squared = 0.534 (+0.534)

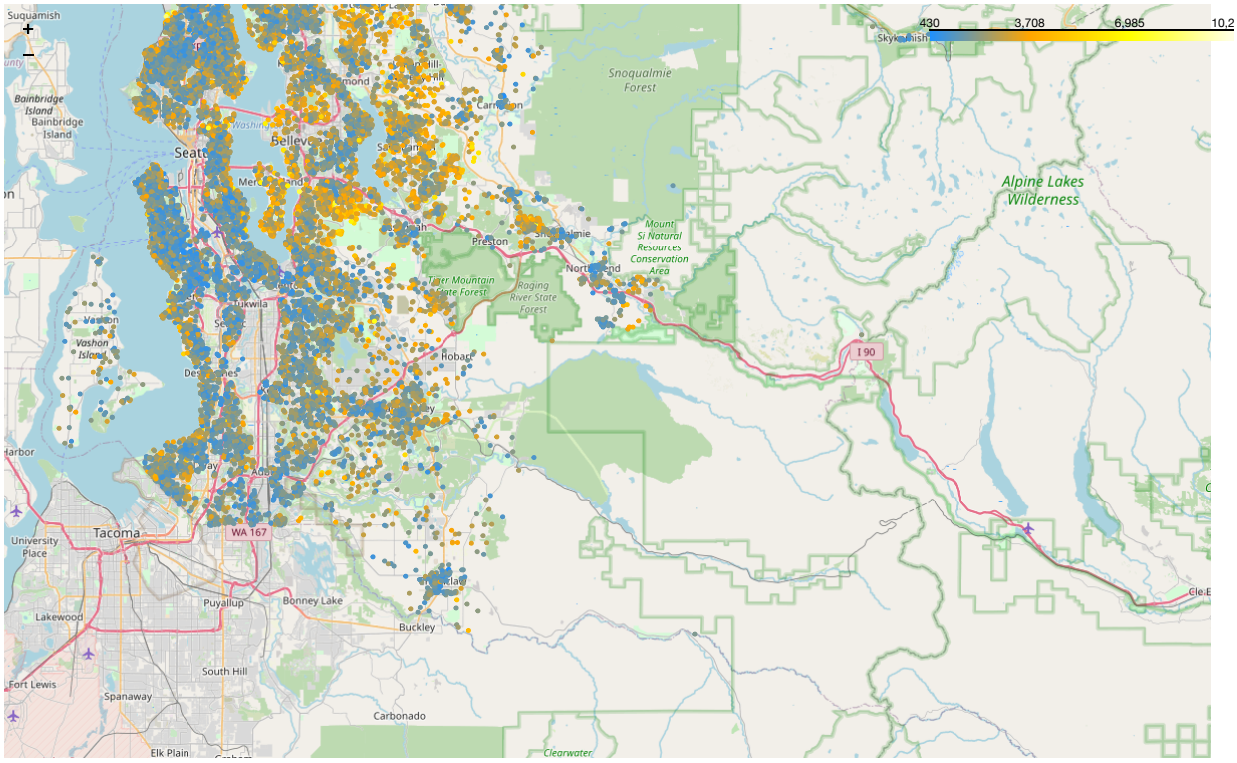
In [49]:
```
# create a LinearColorMap and assign color gradient -> SQFT LIVING
colormap = cm.LinearColormap(colors=['dodgerblue', 'orange', 'yellow', 'lightyellow', 'snow'], vmin=df['sqft_living'].min(), vmax=df['sqft_living']

# create our map again.  This time using a different tileset for a new look
sqft_map = folium.Map(location=center, zoom_start=10)

# Same as before... go through each home in set, make circle, and add to map.
# This time we add a color using price and the colormap object
for i in range(len(df)):
    folium.Circle(
        location=[df.iloc[i]['lat'], df.iloc[i]['long']],
        radius=10,
        fill=True,
        color=colormap(df.iloc[i]['sqft_living']),
        fill_opacity=0.2
    ).add_to(sqft_map)

# the following line adds the scale directly to our map
sqft_map.add_child(colormap)
```

Out[49]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.open

In [50]:
```
sqft_map.save('./visualizations/sqft_map.png')
```

Similarly, for square-footage of living space, we can see that the typical amount is towards the lower end of the spectrum. Within this range, however, there is a strong correlation between the squar and the price of a home. As the amount of living space increases, price tends to increase.

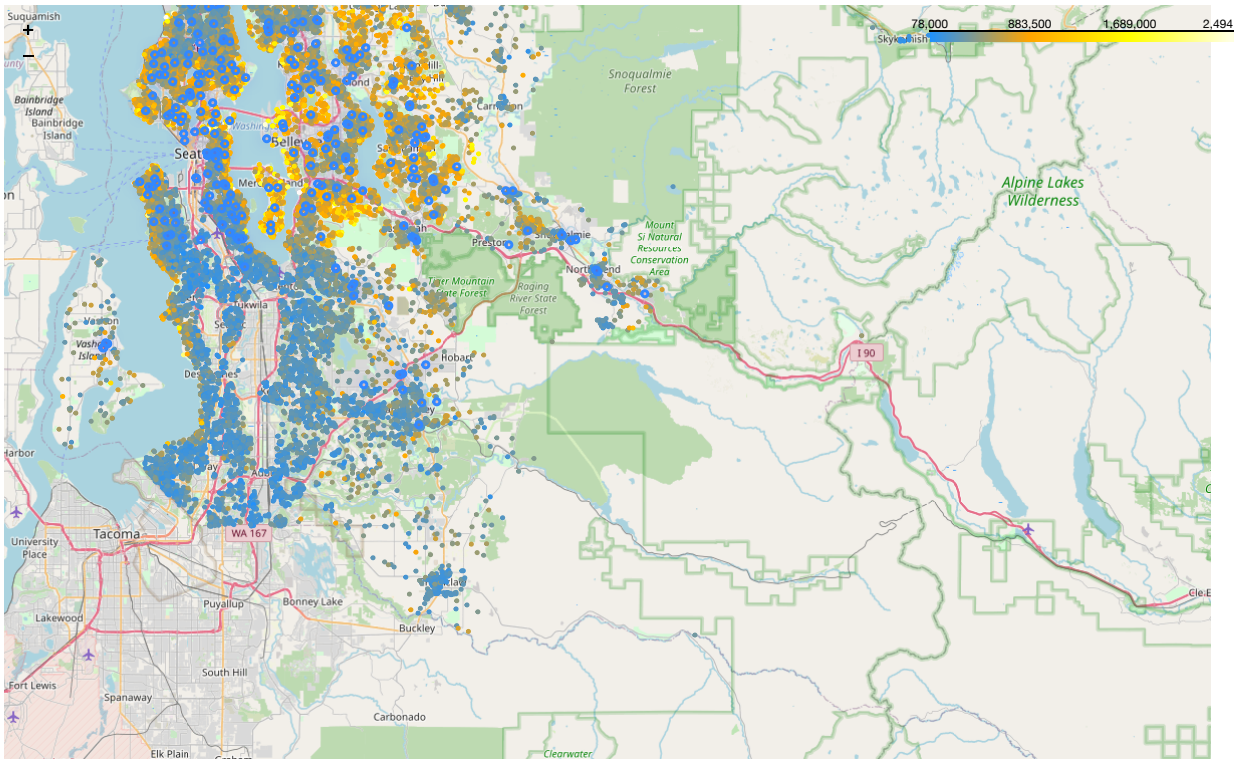
Top Schools | R-squared = 0.672 (+0.138)

In [51]:
```
#plotting the top schools
for index, row in top_schools_df.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius=3,
                        popup=row['school_name'],
                        fill_color='red', fill_opacity=0.1,
                       ).add_to(le_map)
```

In [52]:
```
le_matrix = top_schools_df[['latitude', 'longitude']].as_matrix()

# plot heatmap
le_map.add_children(plugins.HeatMap(le_matrix, radius=3))
le_map
```

Out[52]:



In [53]:
```
le_map.save('./visualizations/school_map.png')
```

The top eight school districts in King County, according to Niche.com, are Mercer Island, Bellevue, Lake Washington, Issaquah, Tahoma, Shoreline, Vashon Island, Snoqualmie Valley, and Seattle Pu blue dots above indicate the location of a school within one of these top eight school districts, and we can expect to see higher-priced homes in their vicinity.

Grade | R-squared = 0.718 (+0.046)

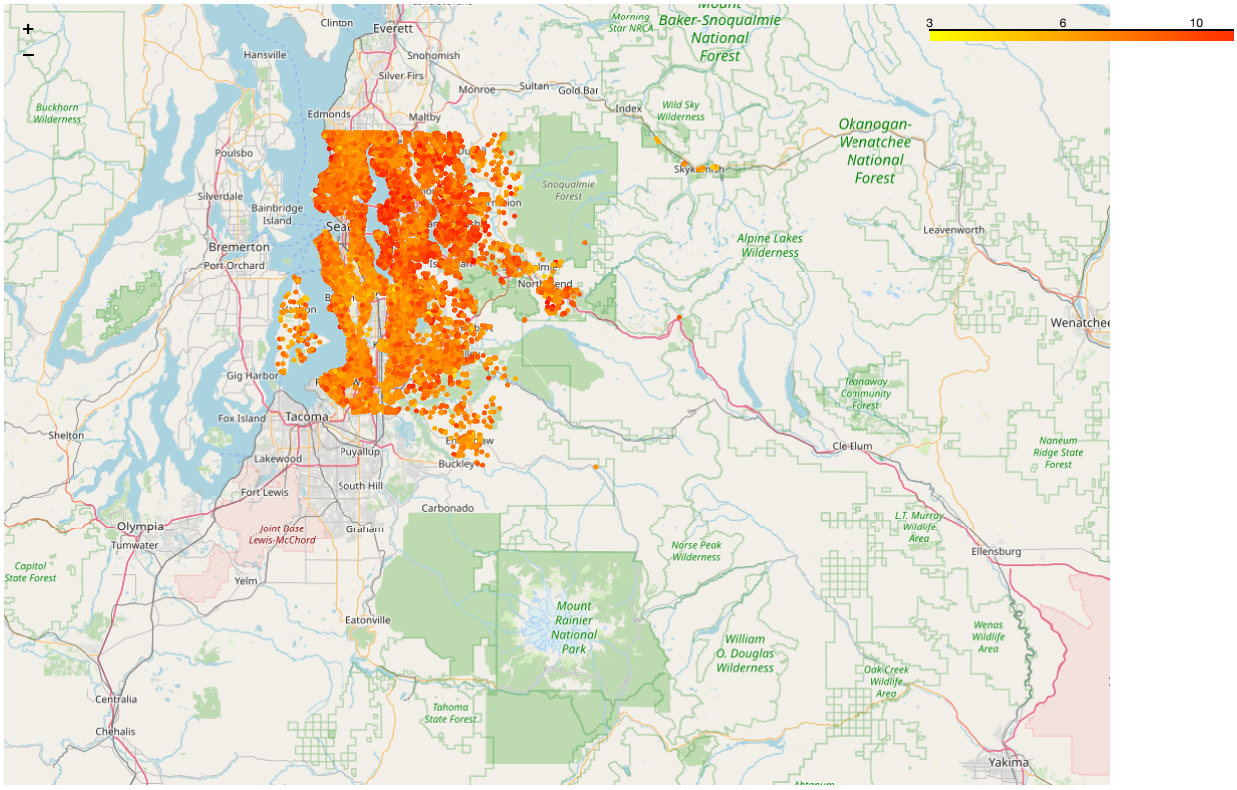
In [54]:
```
# create a LinearColorMap and assign color gradient -> GRADE
colormap = cm.LinearColormap(colors=['yellow', 'orange', 'orangered', 'red'], vmin=df_orig['grade'].min(), vmax=df_orig['grade'].max())

# create our map again.  This time using a different tileset for a new look
grade_map = folium.Map(location=center, zoom_start=9)

# Same as before... go through each home in set, make circle, and add to map.
# This time we add a color using price and the colormap object
for i in range(len(df_orig)):
    folium.Circle(
        location=[df_orig.iloc[i]['lat'], df_orig.iloc[i]['long']],
        radius=10,
        fill=True,
        color=colormap(df_orig.iloc[i]['grade']),
        fill_opacity=0.2
    ).add_to(grade_map)

# the following line adds the scale directly to our map
grade_map.add_child(colormap)
```

Out[54]:



In [55]:
```
grade_map.save('./visualizations/grade_map.png')
```

Here, we can see that it is much more common for the building grade of a home to be on the higher end of the spectrum than the lower, with most houses falling in the middle-upper range. We can
priced homes as the grade incrases.

Church of Scientology | R-squared = 0.753 (+0.035)

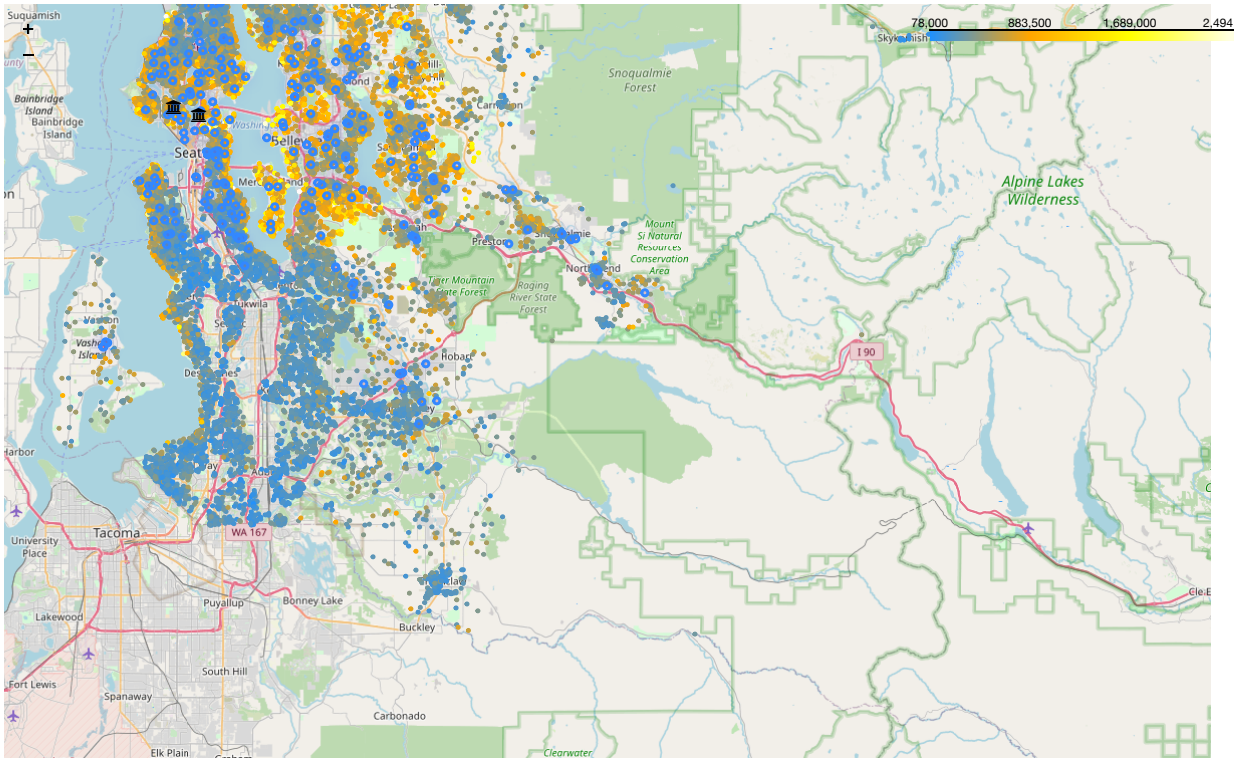
In [56]:
```
#locations pulled from scientology-seattle.org
church_of_scientology_mission = 47.818100, -122.315430
church_of_scientology_washington = 47.622380, -122.361020
church_of_scientology_life_improvement_center = 47.615060, -122.327580
```

In [57]:
```
#Marking the churches of scientology
folium.Marker(church_of_scientology_washington,
              popup='Church of Scientology, Washington',
              icon=folium.Icon(color='pink', icon='university', prefix='fa')
             ).add_to(le_map)
folium.Marker(church_of_scientology_mission,
              popup='Church of Scientology, Mission',
              icon=folium.Icon(color='pink',icon='university', prefix='fa')
             ).add_to(le_map)

folium.Marker(church_of_scientology_life_improvement_center,
              popup='Church of Scientology, Life Improvement Center',
              icon=folium.Icon(color='pink',icon='university', prefix='fa')
             ).add_to(le_map)

le_map.add_child(folium.ClickForMarker(popup="Recommendation for House Placement"))
```

Out[57]:



In [58]:
```
le_map.save('./visualizations/all_map.png')
```

The three pink markers designate scientology churches, around which we can expect to see higher-priced homes.

Bar Plot: Impact of All Features

In [59]:

```
x = ['Square Foot Living', '+Top Schools', '+Grade', '+Scientology', '+Coffee', '+Interaction']
y = [53.4, 67.2, 71.8, 75.3, 76.0, 76.1]
plt.figure(figsize=(14,8))
sns.set(font_scale=1.2)
pal = sns.color_palette("husl", 8)
ax = sns.barplot(x, y)
ax.set_title('Feature vs. Percent Accuracy Model can Predict Price', fontsize=25)
ax.set_xlabel('Additional Feature', fontsize=20)
ax.set_ylabel('Percent Model is Captured by Data (R-squared)', fontsize=20)
ax.text(.115, .565, '53.4%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.text(.28, .705, '67.2%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.text(.445, .753, '71.8%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.text(.61, .79, '75.3%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.text(.78, .795, '76.0%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.text(.945, .796, '76.1%', color='black',
        horizontalalignment='right',
        verticalalignment='top',
        transform=ax.transAxes)
ax.set_ylim(bottom=0, top=100)
ax.set_yticks((0,10,20,30,40,50,60,70,80,90,100))
ax.set_yticklabels((0,10,20,30,40,50,60,70,80,90,100));
plt.savefig('./visualizations/features_barplot.png')
```