Housing Market Prediction Model

November 25, 2024

Rebecca Skinner, Savanna Benn, Harriet Orleans, Lisa Miller

Project objective:

Build a predictive model for home sale prices

Overview

Dataset : Extracted from Realtor.com (2016–2024).

Key Features :

- Month and year
- County and state
- Average listing price
- Active listing count

Deliverables : Cleaned datasets and a trained prediction model.

Data Processing Steps

Data Cleaning

Imported raw dataset using Pandas.

Dropped:

- Columns with >50% missing data.
- Outliers flagged with a quality flag.
- Irrelevant columns, e.g., redundant price data and monthly trends.

Handled missing values and renamed columns for clarity.
Exported cleaned data for further use.

Feature Transformation

- Split `Year/Month` column into separate `Year` and `Month` columns.
- Extracted `State` and `County` from the combined location column.
- Normalized numeric columns using **StandardScaler** for consistency.

Our ETL Process

In [1]:
```
import pandas as pd
from pathlib import Path
```

In [2]:
```
# Store filepath in a variable
housing_csv_path = Path("Resources/RDC_Inventory_Core_Metrics_County_History.csv")
```

In [3]:
```
# Read our data file with the Pandas library
df = pd.read_csv(housing_csv_path)
```

In [4]:
```
# Show the first five rows.
df.head()
```

Out[4]:

	month_date_yyyymm	county_fips	county_name	median_listing_price	median_listing_price_mm	median_listing_price_yy	active_listing_
0	202410	18049	fulton, in	277500.0	-0.1175	0.0282	
1	202410	13027	brooks, ga	259900.0	-0.0048	0.1631	
2	202410	20171	scott, ks	307000.0	0.2280	0.5049	
3	202410	25027	worcester, ma	546175.0	0.0116	0.0762	1
4	202410	18115	ohio, in	288650.0	-0.0503	-0.3349	

5 rows × 40 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310177 entries, 0 to 310176
Data columns (total 40 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   month_date_yyyymm                       310177 non-null  int64
 1   county_fips                             310177 non-null  int64
 2   county_name                             310177 non-null  object
```

Initial look at dataset.

Out[41]:

Index	Year	Month	State	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count	Median Listing Price	Price Increased Count	Price Reduced Count
0	2024.0	10.0	in	45.0	54.0	14.0	146.0	2102.0	363246.0	61.0	277500.0	0.0	14.0
1	2024.0	10.0	ga	25.0	60.0	4.0	139.0	1729.0	295232.0	26.0	259900.0	0.0	4.0
2	2024.0	10.0	va	18.0	28.0	8.0	169.0	1764.0	379463.0	25.0	546175.0	20.0	462.0
3	2024.0	10.0	ga	9.0	46.0	4.0	183.0	1925.0	618415.0	11.0	313950.0	0.0	8.0
4	2024.0	10.0	tx	623.0	67.0	160.0	155.0	1837.0	339294.0	850.0	671725.0	212.0	794.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
144671	2016.0	7.0	ks	60.0	111.0	8.0	60.0	1492.0	104192.0	61.0	349900.0	6.0	246.0
144672	2016.0	7.0	ct	1021.0	64.0	232.0	139.0	1844.0	287802.0	1261.0	74975.0	0.0	10.0
144673	2016.0	7.0	ut	1094.0	68.0	252.0	418.0	2651.0	1888026.0	1384.0	164175.0	2.0	68.0
144674	2016.0	7.0	va	184.0	73.0	52.0	97.0	1907.0	244668.0	240.0	275000.0	4.0	212.0
144675	2016.0	7.0	ky	30.0	114.0	8.0	70.0	1650.0	214186.0	42.0	179200.0	4.0	90.0

144676 rows × 13 columns

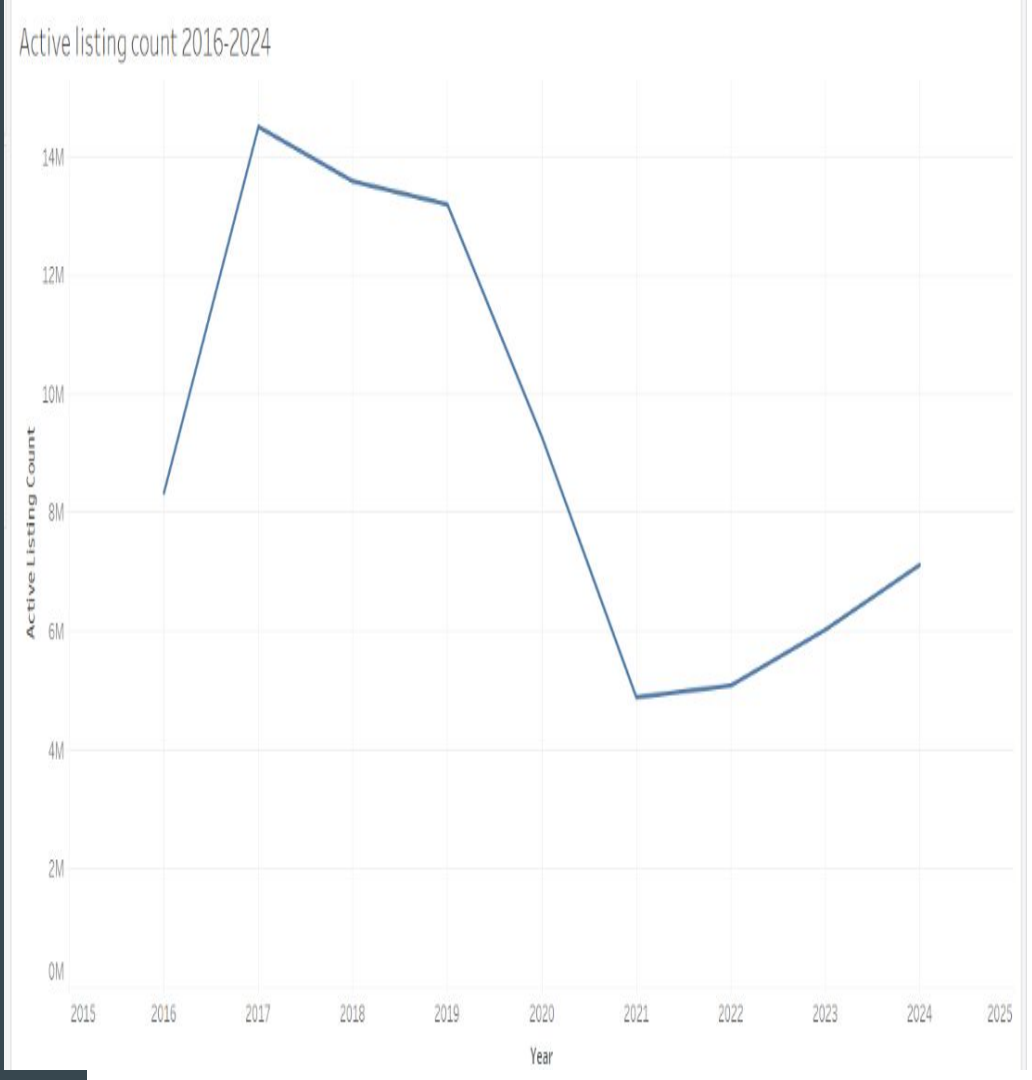
In [42]:
```
# Export csv
opt_3_df_final.to_csv('Resources/opt_3.csv', sep=',', index=True)
```

Cleaned dataset.

Understanding the market

Trend 1

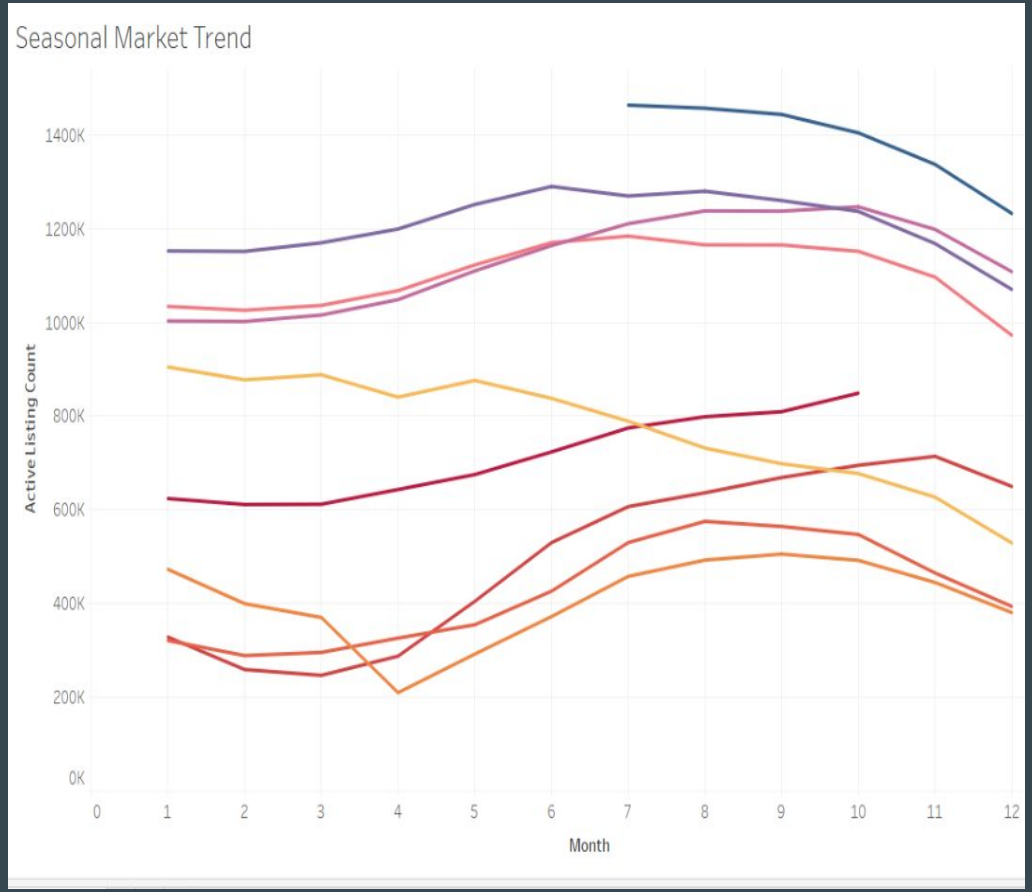
The homes being listed for sale greatly declined beginning in 2019. You can see that covid greatly affected the market. As of 2023 it shows the housing market is recovering and there are more new homes being listed.



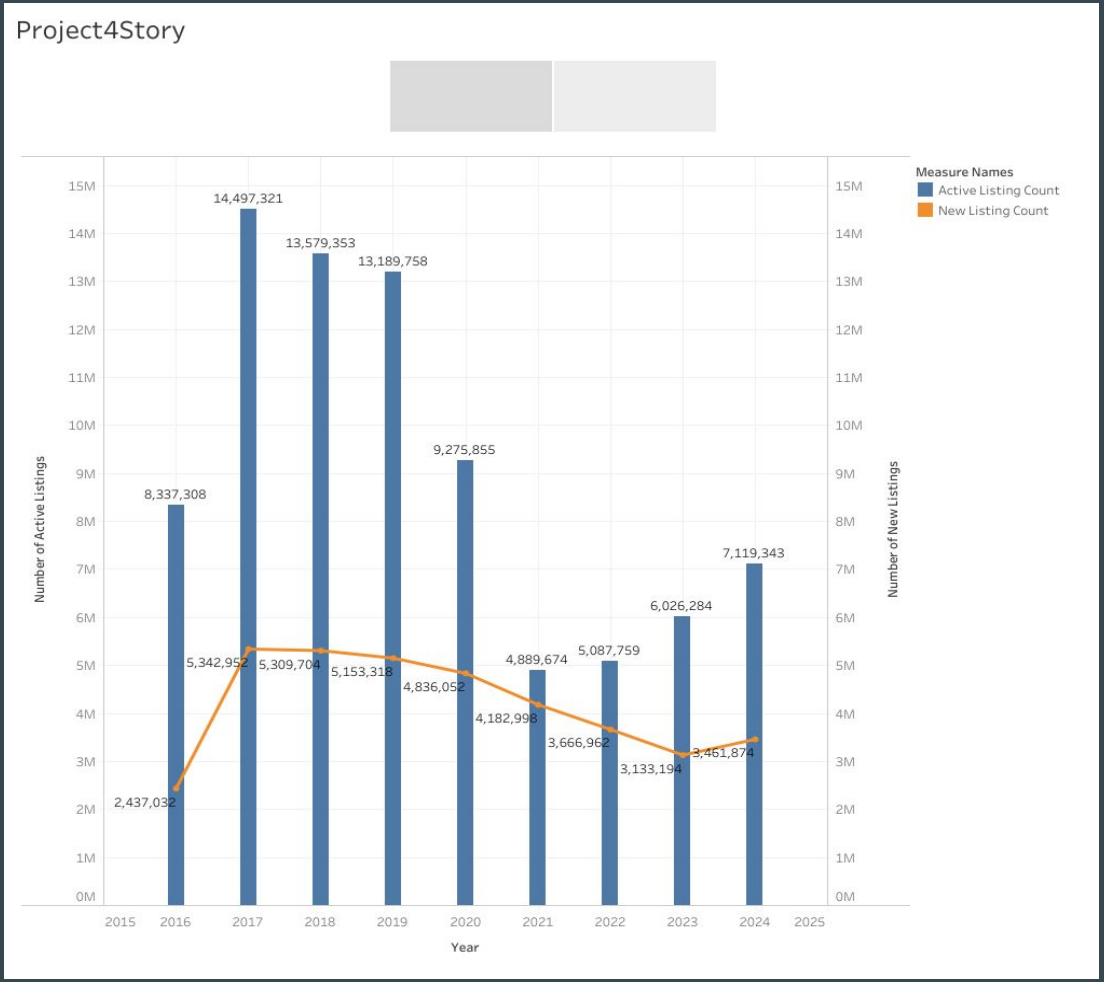
Market trends

Trend 2

Homes are being listed the least in the winter months. Late summer through fall is when the housing market is the busiest.



Listings by Year



Model Training

Tools and Libraries

Programming Languages: Python.

Libraries: Pandas, SQLAlchemy, Scikit-learn.

Database: SQLite for data storage and querying.

Process

Training Data: Split cleaned data into training and testing sets (80:20).

Model Selection: Linear Regression model.

Features Used:

- Median Sqft, Median List Price Per Sqft, Year.
- Encoded state/county as dummy variables for categorical representation.

Model Fit: Trained the model with combined state data.

Model Training

In [2]:
```
import pandas as pd
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from sqlalchemy import text
from sqlalchemy import create_engine, func
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import hvplot.pandas
# Create an engine
engine = create_engine('sqlite:///mydatabase.db')

# Read the CSV file into a DataFrame
df = pd.read_csv('Resources/cleaned_housing_market_data.csv')

# Load the DataFrame into the database
df.to_sql('Housing_Data', con=engine, if_exists='replace', index=False)
Session = sessionmaker(bind=engine)
session = Session()

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 100)
```

In [3]:
```
query_1 = 'SELECT * FROM Housing_Data'
```

In [4]:
```
result = session.execute(text(query_1))

df_result= pd.DataFrame(result)
df_result.head()
#for row in result:
 #   print(row)
```

Out[4]:

	Index	Year	Month	County Name	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count
0	0	2024	10	fulton, in	45.0	54.0	14.0	146.0	2102.0	363246.0	61.0
1	1	2024	10	brooks, ga	25.0	60.0	4.0	139.0	1729.0	295232.0	26.0
2	2	2024	10	worcester, ma	1104.0	31.0	874.0	286.0	1985.0	635754.0	1361.0
3	3	2024	10	sussex, va	18.0	28.0	8.0	169.0	1764.0	379463.0	25.0
4	4	2024	10	clark, wa	1474.0	58.0	604.0	309.0	2225.0	847493.0	2292.0

In [5]:
```
#Scaling Data
# Scaling the numeric columns
housing_data_scaled = StandardScaler().fit_transform(df_result[["Active Listing Count","Median Days on Market","New Listing
                                                                "Total Listing Count"]])

# Creating a DataFrame with with the scaled data
df_housing_transformed = pd.DataFrame(housing_data_scaled, columns=["Active Listing Count","Median Days on Market","New Lis
                                                                "Total Listing Count"])

# Display sample data
df_housing_transformed.head()
```

Out[5]:

	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count
0	-0.339173	-0.566026	-0.344828	0.040579	0.550826	-0.020479	-0.340977
1	-0.359953	-0.378974	-0.366936	-0.026694	-0.423458	-0.214969	-0.365619
2	0.761115	-1.283059	1.556485	1.386050	0.245219	0.758776	0.574306
3	-0.367226	-1.376586	-0.358093	0.261621	-0.332037	0.025895	-0.366324
4	1.145540	-0.441324	0.959561	1.607092	0.872104	1.364258	1.229789

In [6]:
```
#encoding with get dummies
county_encoded = pd.get_dummies(df_result["County Name"], columns=['County Name'])
county_encoded
```

Out[6]:

	abbeville, sc	acadia, la	accomack, va	ada, id	adair, ia	adair, ky	adair, mo	adair, ok	adams, co	adams, ia	adams, id	adams, il	adams, in	adams, ms	adams, nd	a
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	

In [11]:
```
# Make predictions using the X set
predicted_y_values = model.predict(X)
```

In [12]:
```
# Create a copy of the original data
df_predicted = combined_df.copy()

# Add a column with the predicted salary values
df_predicted["Avg_median_predicted"] = predicted_y_values

# Display sample data
df_predicted.head()
```

Out[12]:

	Year	Month	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count	Avg_median_predicted
0	2024	10	-0.339173	-0.566026	-0.344828	0.040579	0.550826	-0.020479	-0.340977	0.102844
1	2024	10	-0.359953	-0.378974	-0.366936	-0.026694	-0.423458	-0.214969	-0.365619	-0.127691
2	2024	10	0.761115	-1.283059	1.556485	1.386050	0.245219	0.758776	0.574306	1.168288
3	2024	10	-0.367226	-1.376586	-0.358093	0.261621	-0.332037	0.025895	-0.366324	0.128726
4	2024	10	1.145540	-0.441324	0.959561	1.607092	0.872104	1.364258	1.229789	1.464613

In [13]:
```
#Formula for the case of three variables.
#print(f"Model's formula: y = {model.intercept_} + {model.coef_[0]}x_1 + {model.coef_[1]}x_2 + {model.coef_[2]}x_3")
```

In [14]:
```
from sklearn.metrics import r2_score
```

In [15]:
```
r2_score( combined_df['Avg Listing Price'] ,predicted_y_values)
```

Out[15]:
```
0.7445084964984598
```

Optimization Attempts

Opt. Attempt 3

	Year	Month	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count	ak	al	ar	az	ca
0	2024	10	-0.339173	-0.566026	-0.344828	0.040579	0.550826	-0.020479	-0.340977	False	False	False	False	False
1	2024	10	-0.359953	-0.378974	-0.366936	-0.026694	-0.423458	-0.214969	-0.365619	False	False	False	False	False
2	2024	10	0.761115	-1.283059	1.556485	1.386050	0.245219	0.758776	0.574306	False	False	False	False	False
3	2024	10	-0.367226	-1.376586	-0.358093	0.261621	-0.332037	0.025895	-0.366324	False	False	False	False	False
4	2024	10	1.145540	-0.441324	0.959561	1.607092	0.872104	1.364258	1.229789	False	False	False	False	False

```
r2_score( combined_df['Avg Listing Price'] ,predicted_y_values)
```

```
0.7526890358540986
```

Optimization Attempts

Opt. Attempt 3

	Year	Month	Active Listing Count	Median Days on Market	New Listing Count	Median List Price Per Sqft	Median Sqft	Avg Listing Price	Total Listing Count	Median Listing Price	Price Increased Count	Price Reduced Count	ak	al	ar	az
0	2024	10	-0.340530	-0.591155	-0.341352	0.053791	0.549356	-0.009965	-0.339564	-0.098480	-0.259439	-0.276710	False	False	False	False
1	2024	10	-0.361033	-0.405071	-0.363447	-0.014468	-0.420277	-0.201677	-0.364025	-0.177867	-0.259439	-0.312173	False	False	False	False
2	2024	10	-0.368209	-1.397520	-0.354609	0.278072	-0.329293	0.035746	-0.364724	1.113402	0.237156	1.312027	False	False	False	False
3	2024	10	-0.377435	-0.839268	-0.363447	0.414591	0.089235	0.709283	-0.374508	0.065931	-0.259439	-0.297988	False	False	False	False
4	2024	10	0.252013	-0.187973	-0.018768	0.141553	-0.139525	-0.077479	0.211848	1.679706	5.004470	2.489395	False	False	False	False

```
r2_score( combined_df['Avg Listing Price'] ,predicted_y_values)
```

```
0.7436066522554001
```

Results

- Initial Results: R^2 score: **0.74**
- Optimization 1 - State Data: R^2 score: **0.75**
- Optimization 2 - Added columns: R^2 score: **0.74**

Summary

Overall, we found the second model, that added in the states column, was the best model for predicting housing prices with an r2 score of 75%. We believe there needs to be a bit more data to achieve a higher score. We also believe having exact housing prices instead of the averages would help our model achieve a higher r2 score.

```
r2_score( combined_df['Avg Listing Price'] ,predicted_y_values)
```

```
0.7526890358540986
```

Resources

Dataset provided by https://www.realtor.com/.

For visualizations published to the public. (version) Tableau 2024.3.1 https://public.tableau.com/views/Project_4_TPW/Project4Story?:language=en-US&publish=yes&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link

Resources provided by instructor assistance, class material, instructor video recordings, and examples given from TTC Bootcamp. Xpert Learning Assistant provided by TCC Bootcamp. Tutoring provided by TTC Bootcamp.